TOOLS FOR THE CONCEPTUAL DESIGN
AND
ENGINEERING ANALYSIS OF MICRO AIR VEHICLES

THESIS

Mustafa Turan, 1st Lieutenant, TUAF

AFIT/GAE/ENY/09-M19

DEPARTMENT OF THE AIR FORCE

AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright Patterson Air Force Base, Ohio

AFIT/GAE/ENY/09-M19

TOOLS FOR THE CONCEPTUAL DESIGN AND ENGINEERING ANALYSIS OF MICRO AIR VEHICLES

THESIS

Presented to the Faculty of the
Department of Aeronautical and Astronautical Engineering
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
in Partial Fulfillment of the Requirements for the
Degree of Master of Science in Aeronautical Engineering

Mustafa Turan
1st Lieutenant, TUAF

March, 2009

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

AFIT/GAE/ENY/09-M19

TOOLS FOR THE CONCEPTUAL DESIGN AND ENGINEERING ANALYSIS OF MICRO AIR VEHICLES

Mustafa Turan

1st Lieutenant, TUAF

Approved:

//signed//	5 Mar 09
Dr. Robert A. Canfield Thesis Co-Advisor	Date
//signed//	5 Mar 09
Lt.Col. Frederick G. Harmon Thesis Co-Advisor	Date
//signed//	5 Mar 09
Lt.Col. Christopher Shearer Committee Member	Date

AFIT/GAE/ENY/09-M19

Abstract

Micro Air Vehicles (MAV) are a subset of Unmanned Aircraft Systems (UAS) that are up to two orders of magnitude smaller than manned systems. Near-Earth environments, such as forests, caves, tunnels and urban structures make reconnaissance, surveillance and search-and-rescue missions difficult and dangerous to accomplish. Therefore, MAVs are considered ideal for these types of missions. Advances in material sciences, analytical tools, propulsion systems, battery technology, etc. have enabled highly effective small-sized aircraft like UAS. Nevertheless, UAS data are not scalable for MAVs due to lack of adequate prior research. While other agencies, universities and some curious hobbyists have done substantial research on this topic, future military missions require MAV designs that meet strict operational performance (range, payload, maneuverability, etc.). Data using full size aircraft is inadequate to characterize miniature aircraft parameters due to the lower Reynolds numbers, low aspect ratio (LAR) wings, and impact of wing-propeller interactions. The main objectives of this research were to: collect and synthesize the available data/tools; create a statistically integrated database/tool set of MAV designs for conceptual design trades; validate the tool set using published experimental data; synthesize and model a prototype design using conceptual and empirical analysis; highlight MAV-specific design criteria; and identify gaps in existing data for later research. The following design tools have constituted the starting point for creating a demonstration tool-set for MAV design: Digital DATCOM supplemented with experimental data (aerodynamics, stability and control), Athena Vortex Lattice (AVL) Method (aerodynamics, stability and control), QPROP (propeller, motor, energy requirement), MATLAB (modeling, aerodynamic equations evaluation, data acquisition, database creation), Microsoft Excel (power/battery modeling) and Phoenix Integration ModelCenter (MC) as the executive control program (integration, siz-

ing and trade studies). Validation cases were completed for the current level of the single-prop, fixed-wing design tool. A coaxial MAV prototype was evaluated and some parametric studies were conducted for QPROP performance.

Acknowledgements

"To invent an airplane is nothing. To build one is something. But to fly is everything." ***Otto Lilienthal***

First, I would like to express my sincere appreciation to my wife for her patience, understanding, gracefully enduring this lengthly process, and providing the needed support all the time. Without her, the stresses and difficulties encountered during this journey would have been felt in much greater amount. I would like to extend my sincere gratitude to Dr. Robert A. Canfield who provided an excellent guidance and continuous support when needed on my thesis work. Lt.Col. Frederick G. Harmon is one of the greatest contributers to my work and also deserves many thanks for encouraging and supporting me throughout the time he was co-advising. I learned a variety of good things from each of them. The remaining committee member, Lt.Col. Christopher Shearer also deserve recognition for his contributions and advices to my work. I would also like to thank to Dr. Michael V. Ol of Air Force Research Laboratories (AFRL) for proposing the idea and sponsoring my thesis work. Dr. Thomas J. Mueller (Notre Dame) and Dr. Mark Drela (MIT) are to be thanked for providing help and solutions for some of the problems encountered the development of the MAV Conceptual Design and Analysis (MCDA) tool. Final thanks to my friends who have contributed to my work for taking their time and helping me finish this project.

Mustafa Turan

Table of Contents

List of Figures

Figure		Page

Figure Page

List of Symbols

List of Abbreviations

TOOLS FOR THE CONCEPTUAL DESIGN AND ENGINEERING ANALYSIS OF MICRO AIR VEHICLES

1. *Introduction*

1.1 Background and Motivation

Unmanned Aircraft Systems (UAS) have become an integral part of the aerospace community. They have numerous military and civilian applications including surveillance, search and rescue, damage assessment, reconnaissance and tactical attack. Currently, the military uses these vehicles primarily for gathering intelligence, surveillance and reconnaissance. The most notable current UASs used by the military for these purposes are the Predator and Global Hawk. Figure 1.1 illustrates the wide variety of UASs including micro air vehicles (MAVs). MAVs have wingspans of approximately 15 cm (0.15 m or 6 in) or less as compared to the larger UAVs, that can be about 3500 cm (35 m or 115 ft) in span (b) as in the case of the Global Hawk [28]. MAVs would be smaller and cheaper than current UASs and could be used to perform similar missions on a different scale. Some of the MAV mission types as mentioned in reference [46] are:

- Surveillance: Day and night video, infrared images of battle fields
- Detection: The sensing of biological agents, chemical compounds and nuclear materials
- Communication: Communication enhancement in urban areas or other environments for continuous line-of-sight operations

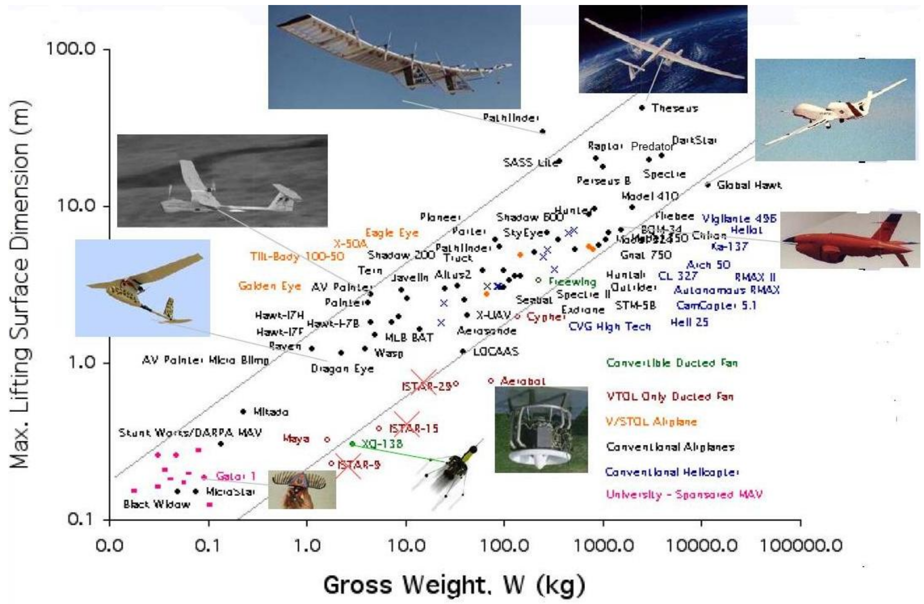


Figure 1.1 UAS Family [10]

- Placement of Unattended Sensors: Acoustic sensors outside of a building for hostage rescue or counter drug operations

Other possible missions are to place acoustic sensors for intelligence or battle damage assessment after attack. These missions may consist of a single MAV or swarming MAVs with multiple sensors. MAVs can fulfill their potential if they attain certain attributes to include: range, endurance, stealth, precision, low cost, low weight, minimal logistical support and mission versatility [28].

In recent years, interest in and development of MAVs has greatly increased. As such, many concepts and designs have emerged for MAVs. However, if one wants to study only certain aspects of MAVs, such as an advanced aerodynamic wing or advanced guidance, navigation and control (GNC), the various designs do not lend themselves well for trade studies. As seen in Figure 1.2, MAVs have a typical Reynolds number (Re) on the order of 5.0x10^4 to 2.5x10^5 and aspect ratios (AR) below two. Often these parameters are driving fundamental research in aerodynamics.

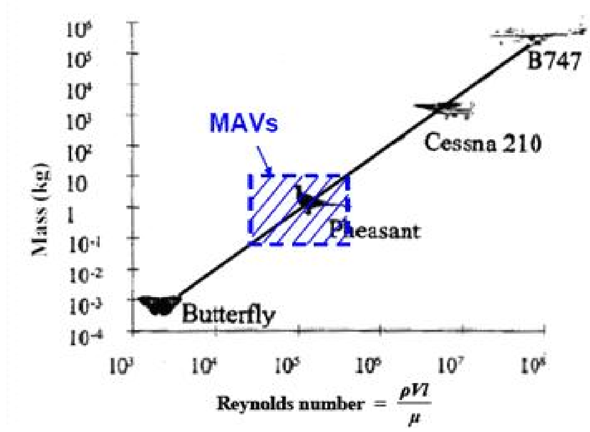


Figure 1.2 Mass versus Reynolds number for MAVs [45]

Conceptual design of aircraft and systems-level analysis for engineering trades all depend on reliable low-level analysis/computation. For manned aircraft and large UASs, aerospace engineering practice benefits from physics-based (lifting-line theory and stability derivatives for airframe performance and flight dynamics analysis, etc.) and non-physics-based tools (extensive statistical databases for table-look up references, etc.). Together these enable quick trade studies and go/no-go evaluations of proposed airplane design concepts.

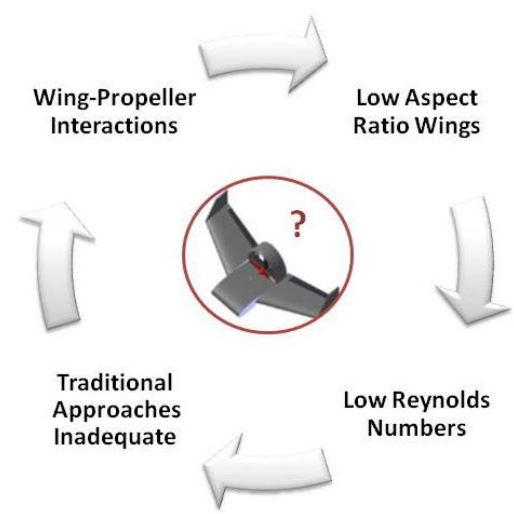


Figure 1.3 Challenges in MAV Design

Figure 1.3 summarizes some of the well-known challenges associated with MAV design. For MAVs, there is a lack of statistical information and considerable doubt on the validity of traditional aerodynamics models. Even for nominally fixed-wing (rigid or flexible-wing) configurations which loosely resemble larger UASs, the combination of low Reynolds numbers, LAR wings and impact of wing-propeller interactions together places traditional models into question. Consequently, the data that we have for full size aircraft do not characterize miniature aircraft well.

1.2 Research Objectives

The main objectives of the proposed research were to integrate a statistical database of MAV designs into look-up tables for conceptual design trades and establish and demonstrate low-fidelity numerical models for MAV aerodynamics and flight dynamics. The current research focuses on collecting and synthesizing the available data and tools, creating a statistically integrated database/tool set, validating the tool set and synthesizing and modeling a prototype design using conceptual and empirical analysis.

1.3 Scope and Assumptions

There are different types of MAVs that are being built by different agencies such as fixed (rigid or flexible-wing), flapping and rotary wing. They all require extensive data and different approaches. This research is focused on rigid-wing MAVs using various tools that have been developed for similar applications. In a similar way to this research, other procedures could be created in order to evaluate fixed (flexible), flapping and rotary wing MAVs as well. Due to the size of the vehicles and the associated Reynolds number, the design of efficient MAVs with classical aerodynamics is questionable or sometimes not possible.

Currently the tool does not include component-level detail, although it can be integrated easily with the background of this research. Also, the final prototype for

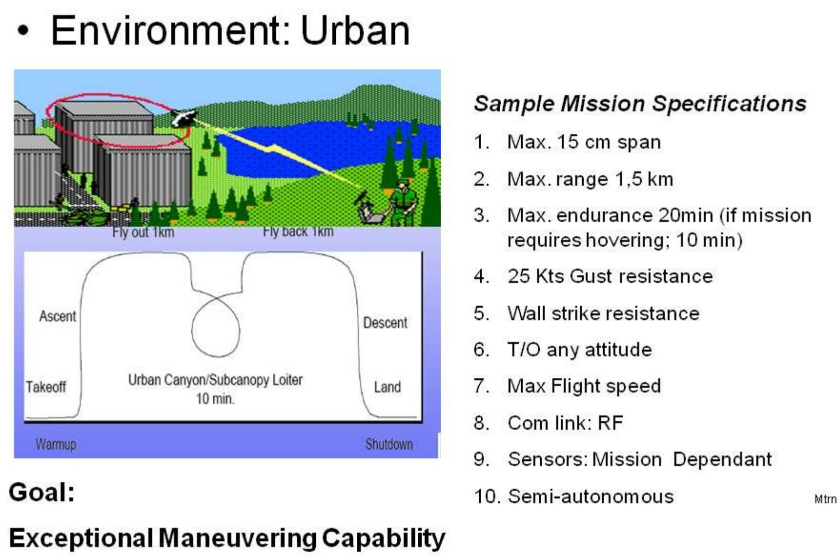


Figure 1.4 Notional Reconnaissance Mission Profile

proposed mission profile (Figure 1.4) is supposed to be a hybrid (flies like a fixed-wing aircraft and hovers like a rotorcraft). Hence, some part of the mission has to include transition from level flight to hover mode or the other way around. The transition part of the flight is beyond the scope of this research.

Some assumptions had to be made due to the specific tools employed or the limited data available. For instance, DATCOM requires certain parameters that go into calculations as inputs. DATCOM handles straight-tapered, cranked or double-delta wings. For Zimmerman, Inverse Zimmerman and elliptical planforms, it is not possible to enter planform properties due to the limitation on inputs. Some measures are taken to alleviate this problem and new geometry inputs are created using some assumptions. The first assumption is that an equivalent trapezoidal wing planform based on AR and b is sufficient and the second one is that the equivalent body based on fineness ratio for the body can be used. Also, the backbone of the experimental data analysis is based on the final report [47] and dissertation [59] of Torres which, though limited, is useful to demonstrate procedures for using experimental data with the following assumption. The third assumption is that in the areas where

DATCOM fails, experimental data can be interpolated if experimental database is created properly. The final assumption is that Drela's QPROP for propulsion and AVL for stability and control can be used to demonstrate a tool set. These assumptions constitute the basis of the methods implemented in this research.

1.4 Hypothesis

Using ModelCenter (MC) [50] as the executive control program to orchestrate other tools of interest, it is possible to develop a conceptual design tool for fixed-wing MAVs and extend the current capabilities further. Also, this research will help people create a database, procedures and templates (for the future use of the MC) for the conceptual design of MAVs. Moreover, the same type of design environment may be extended for evaluating rotary and flapping wing MAVs.

1.5 Methodology

Based on the previously mentioned restrictions in the conceptual MAV design, to extend that data to the design of MAVs, various tools will be integrated into a program called MC. A similar modeling/optimizing environment in MC was applied to a Joined-Wing Sensor-Craft by Dittmar [16] and a successful concept validation model was constructed based on an S-3 Viking with the values within 4% of the actual published aircraft values. For MAVs, a similar environment, but using different tools, will be examined in a multidisdiplinary approach. Mueller's text books [46, 45] on MAVs were used as a starting point for a multidisciplinary approach to a fixed wing MAV design. This research aims to develop procedures for building a fixed-wing MAV conceptual design and analysis tool via a combination of a survey of state of the art and original model/code development. Overall, this research encompasses the aerodynamics, propulsion (propeller-type), stability and control pieces of the conceptual MAV design process in a multidisciplinary approach.

There are three design phases: conceptual, preliminary and detail design [54]. As the first step, the conceptual design process is a very fluid process with many flexibilities. As one goes further in detail, new constraints emerge and the design gets more sophisticated. Aircraft systems are very complex and require intensive multidisciplinary optimization (MDO). Sometimes bringing people from different disciplines together and allowing them to apply their expertise and knowledge on the design of interest can be costly if there is not enough interaction. Software programs such as MC helps engineers share and interact with each other (within the discipline or between disciplines) leading to robust and successful designs throughout the phases of the aircraft design. Following the methodology listed below, a multidisciplinary conceptual MAV design tool was created in the MC environment and it is named as the MAV Conceptual Design and Analysis (MCDA) tool:

1. Considering different design tools, methods and determining the ones applicable to the conceptual design of MAVs
2. Setting up the design environment in a commercial design and optimization software program called MC
3. Incorporating different applicable design tools and experimental results into MC via various interfaces
4. Determining procedures for implementing experimental data into the tool
5. Validating the tool with the published references
6. Evaluating the expected results

The ultimate goal is to have an approach to provide rapid and economical estimation of aerodynamic, propulsion, stability and control characteristics.

1.6 Outline of the Chapters

In Chapter 1, we discussed the background and motivation for this research. The scope of the research and assumptions for related parts of the research were

also discussed. The hypothesis, methodology, and the outline of the chapters can be reviewed in this chapter as well.

Chapter 2 is a review of the different disciplines related to the accomplishment of the current research. Although there are several resources on the matter, it is restricted intentionally to the current research aims.

Chapter 3 primarily focuses on the development of the fixed-wing MCDA tool. Selection methodology of the tools is mentioned in this chapter. The first part includes brief information about the major software programs that have been integrated into the MCDA tool. The second part includes the filewrapper structures of the related tools. The third part has detailed summaries of the components in the MCDA tool. Code, filewrappers and templates are attached in Appendix D.

Chapter 4's first part covers the experimental data interpolation evaluation of the MCDA tool. The current level of the tool is compared with published references. Marek [38] and some of his procedures and a case study in that reference will be evaluated for the validity of the tools and codes in the second part. The third part covers the conceptual design of a coaxial prototype with some modifications to the single-propeller MCDA tool and some analysis of it. In the last part, QPROP performance is evaluated with the MC parametric study tool without validation.

Chapter 5 discusses the conclusions and recommendations. The MCDA tool has some limitations and they will also be presented here. Future advancements and additions to the tool with some recommendations are discussed, concluding the thesis.

Appendices are addressed within the text.

Part of this thesis has been presented in 4^{th} Annual Dayton Engineering Sciences Symposium (DESS08-0065) 27 October 2008 , Dayton, OH and 47^{th} AIAA Aerospace Sciences Meeting (AIAA-2009-38) 5-8 Jan 2009, Orlando, FL.

2. Literature Review by Topic

2.1 Aerodynamics

The wing planform and the airfoil section of the lifting surface are critically important to the performance of all flying vehicles. As in all air vehicles, it is intended to have stable and controllable vehicles. Mueller's textbook [46] covers the discussions on the progression from high Reynolds numbers to the low Reynolds numbers encountered in the design of MAVs. In the same chapter, it is explained that classical aerodynamic theory is not adequate for low Reynolds numbers and LAR wings due to unique aerodynamic properties. MAV applications are hindered by the lack of thorough understanding of the aerodynamics associated with MAVs flying at low speeds. Some of the early experiments on LAR wings at low Reynolds numbers were run between 1930 and late 1950 by various researchers. According to the previous researchers, Mueller [48] and Torres [47, 59]:

- A finite wing of a given AR generates lift from counter-rotating wing tip vortices
- These vortices strengthen as the angle of attack (AoA) increases
- For a LAR wing, wing tip vortices might be present over most of the area
- LAR wings can have linear and non-linear sources of lift. The non-linear lift due to the tip vortices results in an increased lift-curve slope and therefore a high value of α_{stall}

Therefore, some measures had to be taken in order to precisely calculate the effects of LAR wing at low Reynolds numbers. Some suggestions were made by Polhamus [52] and extended by Lamar [36] for non-linear equations for lift, drag and pitching moment. The experimental data in reference [46] were used to calculate the parameters for Pollhamus' lift, Prandtl's drag and Lamar's non-linear pitching moment. It was found that the non-linear equation approximations are only applicable up to α_{stall}.

At greater values of AoA, the highly nonlinear effects associated with pre-stall and stall conditions cannot be modeled by simple equations. Similar wind tunnel tests and procedures were carried out by Marek [38] and the method was validated for higher Reynolds numbers with similar agreement to Mueller and Torres' [47] work. In another publication, Torres and Mueller [60] presented their previous research on LAR wings at low Reynolds number adding Vortex-Lattice Method (VLM) implementation and they compared the experimental results with VLM predictions. For certain cases, the VLM results compared well with experimental data.

The present research mostly focuses on the procedures of those references due to the limited nature of non-linear equation approximations. Although this reason narrows the capabilities of the MCDA tool, it is capable of generating the experimental test results with a good approximation.

One of the main challenges associated with MAV design mentioned in Chapter 1 is the effects of propulsive-induced flow on the aerodynamics of MAVs. Null et al. [49] conducted some experiments to figure out the flow interaction over the aerodynamic surfaces. Some of their findings were:

- The propeller-induced flow had the largest influence on the lowest Reynolds number test cases
- The propeller-induced flow does have a substantial effect on the aerodynamics of the typical MAV where the propeller diameter is a significant portion of the wingspan
- The induced flow from the propulsion system had a positive effect on the lift coefficients (C_L) of the vehicles. The induced flow caused somewhat higher magnitudes of maximum lift coefficient (Cl_{max}) and a delayed stall, but a detrimental effect on the drag coefficients (C_D) and a subsequent decrease in the lift-to-drag (L/D) ratios at low angles of attack. L/D ratios at high angle of attacks were higher for the propulsive-induced tests.

Also in Galinski [20]'s MAV configuration, it was found that propeller-induced flow at the control surfaces acted as an additional advantage almost equivalent to the thrust vectoring of the modern fighter airplane.

2.2 Power/Propulsion

The design of an electric propulsion system for an UAS incorporates various disciplines making the task a MDO case. Propeller aerodynamics, structural properties, characteristics of the electric system and the vehicle itself require great efforts in propulsion system design. Sibilski et al. [56] described power requirements for MAVs. Within the comparison of the three flight modes (fixed, rotary and flapping wing) when there is no hover requirement, fixed wing flight is always the most energy efficient for MAVs. Nevertheless, if there is a hover requirement, flapping or rotary wing would be the preferred options based on the constraints.

In other research, Gur and Rosen [23] presented a comprehensive method for optimal design of electric propulsion systems for UASs that are also applicable to the MAV cases. It is known that the propulsion system of MAVs (batteries, motor, propeller, etc.) accounts close to $60-70\%$ of the vehicles total weight [44]. Therefore, optimization of the propulsion systems is extremely important. Gur and Rosen [24] also studied common models for the analysis of the propeller aerodynamics, performance calculations and propeller design and compared the results, discussing the advantages and disadvantages of each one.

Propeller research at the University of Illinois at Urbana-Champaign included the study of small scale propellers which have been widely used for radio controlled (R/C) aircraft. A large number of off-the-shelf R/C aircraft that have different sizes and shapes of propellers are now available. These aircraft are fairly inexpensive and use small motors and propellers with diameters less than 5 in to provide thrust. An increased interest in MAVs in industry and for the military creates a need for data on micro propellers as well. Deters and Selig [15] conducted some experiments

on different scales of micro propellers in order to measure the static performance. According to them, knowing the static performance of micro propellers would be very useful in determining motor selection and flight capabilities of small R/C aircraft and also MAVs. This type of research is also useful for the QPROP propeller input file which is a part of the MCDA tool.

Mueller et al. [46] mentioned different types of propulsion (internal combustion engine propulsion and electric motor propulsion), DC electric motors (cored, coreless and brushless), batteries (lithium polymer, nickel-cadmium and nickel-metal-hydrite cells) and electric motor controllers (brushed "Electronic Speed Control-ESC", brushless). They explained the advantages and disadvantages of those units in detail for a better MAV design.

2.3 Stability and Control

The first-order derivatives of the aerodynamic coefficients are called stability derivatives. According to Krashanitsa et al. [35], stability and control are of primary concern for MAV design due to beyond line-of-sight operations. Since MAVs demonstrate intrinsically unsteady behavior with high and low frequency oscillations, flying the MAV via an on-board camera and controlling it from a ground station is very difficult. Therefore, they emphasized that an enhanced Automatic Flight Control System (AFCS) is required. Their research covers the methods of development and integration of systems for the autonomous flight of a MAV. Moreover, the process also includes the determination of stability and control derivatives using analytical and numerical computational software, simulation of flight dynamics and closed-loop control design. The linearized equations of motion of the aircraft were used to evaluate the effectiveness of the control laws for MAVs and the equations are developed from an evaluation of the various aerodynamic stability and control derivatives. Aerodynamic derivatives were determined by the use of analytical software. Advanced Aerodynamic Analysis (AAA) software, VLM and Tornado software

by Melin [42] were used in determining the stability and control derivatives. But one thing to mention here is that the propeller moments and propeller wash on the wing had been neglected. Dragonfly and Zagi MAVs were used for flight testing in their research.

Melin [41] worked on a MATLAB code to evaluate the aerodynamic properties using VLM and developed a vortex lattice code "Tornado solver". In order to validate Tornado's data, he compared the results with two different methods: VLM and panel code. AVL, VIRGIT and Tornado are all vortex lattice methods while CMARC is a panel method. He worked on a large-scale "Cessna 172" but was able to get Tornado computational results for the Cessna 172 that correlated with both AVL data and CMARC data. His approach and the results are explored in his thesis [41] and manual [42].

In addition to those efforts, Kellogg [33] from the Naval Research Laboratory (NRL) summarized some results for stability and control behaviors of the micro tactical expendable (MITE) series. Another effort on the same topic by Stewart et al. [58] included a MAV configuration with flexible wings and they aimed to look at the issue of air vehicle flexibility on the flight mechanics and also control aspects particular to MAVs. They estimated the airframe aerodynamic coefficients using AVL and compared to wind tunnel data to ensure the estimates from AVL were reasonable. Some of the comparison figures are presented in their paper. They also evaluated the stability and control characteristics of their MAV.

Digital DATCOM [39] provides the longitudinal coefficients and the derivatives of them for static stability characteristics. Output for configurations with a wing and horizontal tail also includes downwash and the local dynamic pressure ratio in the region of the tail. Also, dynamic stability characteristics can be computed for each component with some limitations. Whether they can be used for MAVs or not is not clear since the program was designed for normal size aircraft. DATCOM also

has capability to evaluate the stability and control characteristics as mentioned in the manual [39] and might be appropriate to evaluate MAVs.

2.4 MAV Design Efforts

A large number of MAVs have been designed and flown around the world for different purposes. Several universities have been involved in MAV research. Pines and Bohorquez [51] gathered information on the notable firsts in the MAV Flight Regime. The list follows as in [51]:

1. The first battery-powered electric motor open-loop controlled flapping flight was by Microbat-CalTech/Aerovironment.
2. The longest endurance (< 100 g) is > 30 min for a fixed-wing MAV by a Black Widow designed and built by Aerovironment.
3. The first autonomous MAV flight (global-positioning-system waypoint navigation) was by Microstar-Lockheed Martin.
4. The first open-loop controlled hovering flight of a biologically-inspired flapping vehicle was by MENTOR-SRI.
5. The longest-endurance flapping flight (< 100 g) was about 25 mins by a 9 in Microbat designed and built by Aerovironment.

This list will grow whenever new technologies emerge such as micro fuel cells or motors and the physics is discovered behind some of the challenges of MAV development. MAV design has become a catalyst for research in aerodynamics, propulsion, stability and control, MDO, microelectronics and artificial intelligence.

Figure 2.1 illustrates the different types of MAVs: fixed-wing, flapping-wing, rotorcraft and hybrid (tailsitter or tilt-rotor). Meuller's textbook [46] is a great reference for MAV designers and has three case studies although limited to certain types of MAVs. Torres [59] provided comprehensive aerodynamic data and showed how useful engineering decision-making tools could be extracted from wind tunnel

data for MAV design. Marek [38] also conducted some wind tunnel experiments similar to Torres [47] and compared the results' validity and created a database for his design and optimization instead of using the VLM or some sophisticated computational fluid dynamics (CFD) code. Therefore, he used a method based directly on data from wind tunnel experiments. In his wind tunnel experiments, the method was validated for higher Reynolds numbers than described in Torres [47]. He also built and flew the "BumbleeBee" MAV which is one of the validation cases of the current research.

Pines and Bohorquez's paper [51] about the challenges of the future development of MAVs is also a very good reference for different advancements/issues on MAV design. The status and performance of the current MAVs of that time (2006), fundamental physics limiting MAV performance and emerging MAV research and technology trends are explained thoroughly.

Black Widow

Ornitopter

The University of Delaware's Sparrow

The University of Sydney's T-Wing

E-Flite Blade MCX

X4-Flyer is a quadrotor platform

Figure 2.1 MAV Types

Some other MAV development efforts are as follows:

Kellogg [33] from NRL describes the development of the MITE fixed-wing MAVs developed at the NRL from 1996 to 2002. The development of the MITE program is explained in a theoretical, experimental and practical way. Aerodynamic design refinements by CFD, propulsion system and design, structure, flight experiments, design evolution (MITE 1-2-3-4-5-6), stability and flight control and navigation processes are explained with the key points and talent required to build and operate a MAV. In 2003, with all the experience gained from the MITE series, a transition to the first operational electric-powered, back-packable airplane "Dragon Eye" was made successfully, although it was not a MAV with respect to its size. Grasmeyer and Keennon [21] also describe the extensive research on the development of the "Black Widow" and this program showed that a 6-inch MAV can perform useful missions that were deemed impossible previously.

In another research effort, Green [22] shifted the focus to another issue by emphasizing that limited flying space and densely populated obstacle fields requires a vehicle that is capable of hovering as well as being highly maneuverable. He incorporated a secondary flight mode into a fixed-wing aircraft to preserve its maneuverability while adding a hovering capability. For this purpose, he designed a fixed-wing hybrid platform with a high thrust-to-weight ratio (T/W) enabling it to transition into a vertical flight mode and ran some experiments using his approaches for transition. Importantly, he demonstrated that hovering mode can be sustained using the propeller wash and enlarged elevator and rudder control surfaces as seen in Figure 2.2. He also developed a quaternion attitude controller. This approach was successfully applied in his case due to the large AoA maneuvers of the hybrid platform. As a result of his experiments, captured flight data showed that the controller's performance exceeded that of an expert human pilot who would fly the platform on the prop (prop hanging). Another important result is that, while hanging, the motors reactive torque was thought to be beneficial, but the plane torque-rolled at a rate of 20-25 RPM, causing a dizzying effect on the video capture. He mitigated

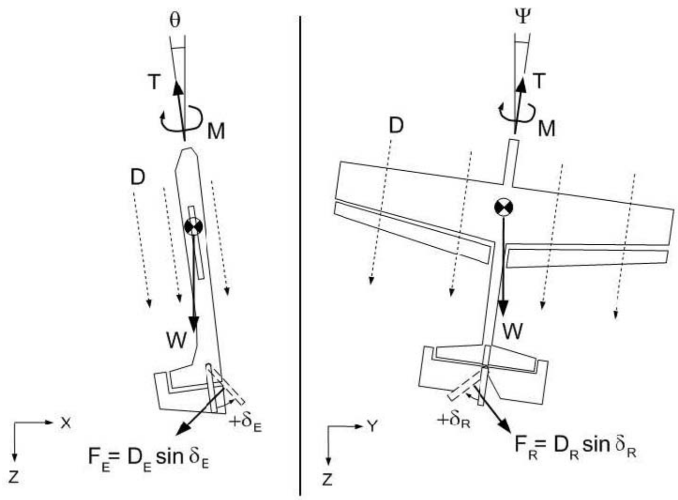


Figure 2.2 Regulating the Pitch and Yaw Angles via Elevator and Rudder [22]

this problem with two DC motors with propellers, mounted on each wing tip and oriented such that the thrust vectors had an angular separation of 180 degrees. This created a counter-rotating force and by controlling the speed of the wingtip motors, the torque roll was regulated. The final system was able to achieve the described mission in his thesis. Another approach to the same issue is coaxial rotors. Successful applications of coaxial rotors as in a Ka-50 helicopter and some UASs increased the interest in that topic. He et al. [26] tested the feasibility of hovering a MAV using a single-motor, double-rotor gearless torque-canceling mechanism without the need for complex gear or electrical control systems. Their results show that such a mechanism is feasible and is able to produce adequate thrust with insignificant net torque on the MAV. However, they found that their prototype did not produce the thrust required to hover their prototype due to an inefficient motor and friction in the slip ring system. In the R/C community, there are several examples of indoor and outdoor coaxial helos such as E-flite Blade MCX. But they use double-motor, double-rotor and variable speed controllers for maneuverability. One successful example of coaxial type propulsion is the "Mini-Vertigo" by Moschetta et al. [43] as seen in Figure 2.3. They tested two fixed-wing MAV configurations: a tilt-wing con-

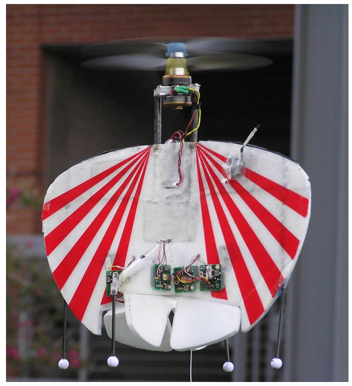


Figure 2.3 Mini-Vertigo in Hover Mode [43]

cept powered by two non-coaxial counter-rotating propellers and a tilt-body concept based on a coaxial rotor. They paid more attention to the coaxial tail-sitter concept for which the propellers induced flow that guaranteed aerodynamic efficiency over the whole flight envelope. They observed that the drag force on the wing is produced by a mixture of free stream and propeller-induced stream. Additionally, the zero-lift drag coefficient increased by about three times with propeller-induced speed increase from 0 to 15 m/s. The flight tests in R/C mode proved that their present configuration had very good handling capabilities for both horizontal flight and hovering with the tilt-body fixed-wing coaxial configurations. The reader may refer to these references [14, 32, 25, 34] for more information related to these topics on coaxial rotor systems.

Other applications related to hovering flight are compound aircraft, tilt-wing, tilt-rotor and tail-sitters. Comparing those different applications, Hogge [27] concluded that a tail-sitter is the best option in terms of being able to perform well both in vertical and horizontal flight. His work includes designing, analysis, constructing and flight testing of some conceptual miniature Vertical Take-Off and Landing (VTOL) tail-sitter UAV prototypes. He also included a method for sizing control surfaces for a tail-sitter vehicle. His approach to the case was:

- A combination of classic aircraft design methods and numerical analysis were used to estimate the aircraft performance and flight characteristics
- The numerical analysis was based on a propeller blade-element theory coupled with momentum equations to predict the influence of a propeller slipstream on the freestream flow field
- Analyzing the aircraft was accomplished using 3D vortex lifting-line theory to model finite wings immersed in the flow field

In order to manage those steps, he utilized a program called "AITHER" developed by Hunsaker [29].

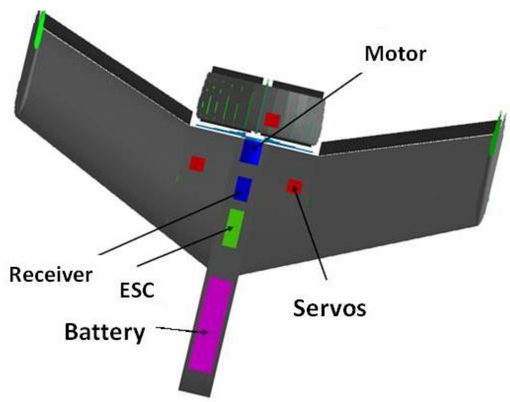


Figure 2.4 Hogge's Final Prototype and Component Placement [27]

Figure 2.4 illustrates the final prototype of Hogge [27]. Development of a control system which was effective for vertical flight while the vehicle was descending, or hovering in ground effect, was achieved. However, developing solutions for obtaining the desired hover flight time was the challenging part of the research. His recom-

mendations on tail-sitter MAVs are: counter-rotating propellers, vectored thrust, variable pitch propellers and improved propulsion technology.

2.5 Software

Digital DATCOM was used in the primary design of aircraft for rapid and economical estimations of aerodynamic stability and control characteristics that are frequently required. The fundamental purpose of the USAF stability and control DATCOM is to provide methods for estimating stability and control characteristics in preliminary design applications. It calculates static stability, high-lift and control device and dynamic-derivative characteristics. Trim option is also available [39]. DATCOM+ is a more user-friendly version, available online with modifications [2].

Many MAV designers use PROFOIL, Eppler Code, Drela's XFOIL, Profili [53] (derived from XFOIL), XFLR5 (derived from XFOIL) or XWING [37] for aerodynamic data. In one research project, Selig et al. [55] compared modern computational tools of the time (PROFOIL, XFOIL, Eppler Code) and wind tunnel tests for low Reynolds number airfoil design and analysis. Among these programs mentioned above, XWING seems to be the only one with 3D effects included. XWING uses a 2D boundary layer and a 3D potential flow matching technique and the capabilities of the new software were examined for swept, twisted, and tapered wings at high Reynolds numbers as well as for LAR-rectangular wings at low Reynolds numbers. The technique was determined to be particularly useful for 3D wings at low Reynolds number [37]. Since most of the applications for fixed wing MAVs utilize LAR wings, it would have been beneficial to integrate this code into MC for a better analysis. However, at the time this research was being conducted, the XWING program was not yet available. It was decided that limited experimental data and DATCOM would be used together for the aerodynamic coefficients which would have 3D effects for the analysis.

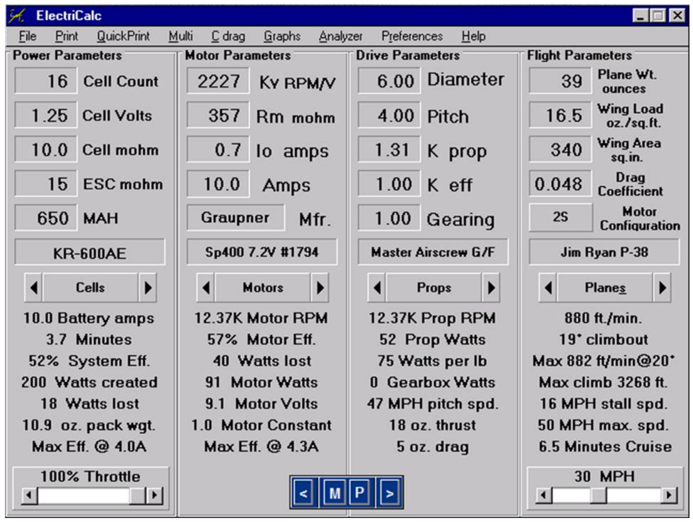


Figure 2.5 ElectriCalc Screenshot

R/C communities use commercial software such as ElectriCalc, MotoCalc and also freeware such as Estimate Electric Motor and Prop Combo or various Excel Spreadsheets. Some of those programs have their own database and are able to calculate flight parameters based on motor, battery, propeller and electronic speed controller as seen in Figure 2.5. They may provide acceptable results for R/C aircraft but the order of magnitude of the sizes of the components that are used in R/C aircraft is still higher than the MAVs and most of them don't take low Reynolds number effects into account. Some of the tools utilized in the current research are Drela's QPROP [17], AVL [19] and performance analysis Excel Spreadsheets. In Appendix A, there is a detailed list of various calculators compiled online [61, 7, 5] which might be useful for follow-on research.

3. Integration Environment

Multidisciplinary system design is a computationally intensive process combining individual disciplines within the design environment. Performing a design and optimization with limited interaction with other branches of research makes the process costly and sometimes inefficient. The present research combines different disciplines into a design and optimization tool called ModelCenter (MC) which will increase the computational power and communication capabilities between researchers during the conceptual design of MAVs. As in Figure 3.1, the following components were intended to be integrated into ModelCenter in order to create the MCDA tool.

- MC (sizing and trade studies)
- Digital DATCOM (aerodynamics, stability and control)
- QPROP (power, thrust, energy requirement)
- AVL (stability and control, aerodynamics)
- MATLAB Plug-in (for various applications)
- Excel Plug-in, eHeli [4] (power/battery modeling)
- ElectriCalc or MotoCalc Database
- Script Program (MC)

In determination of the components to be integrated into MC, the R/C world was explored since the tools and materials utilized in most of the MAV design efforts were from R/C solutions although the size of the R/C aircraft is still larger than MAVs. There are many applications, software and tools available and some are proprietary. Appendix A has a list of some of the tools. Some criteria were considered in selection of the tools:

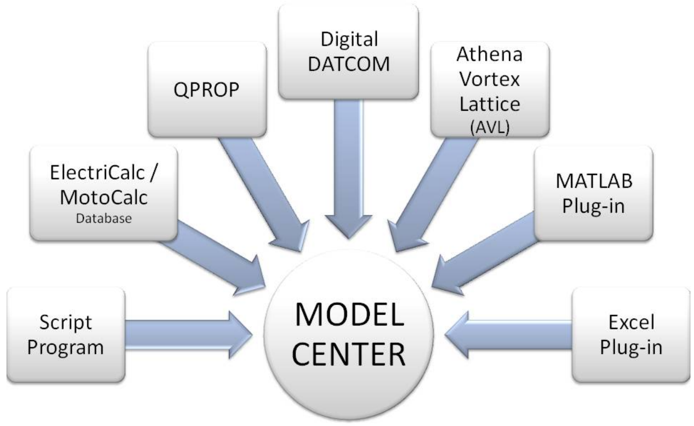


Figure 3.1 Overview of MC Component Integration

- Integratibility: As a first requirement, tools must have the appropriate type or format to be integrated into MC. So any MATLAB, MathCad or Excel file can be an option. Also some I/O programs can be used via filewrappers.
- Documentation: Most tools available online don't have any documentation and a user has to figure out how they work on his own. It causes a loss of time and effort.
- Sufficient Theoretical Background: The solutions must have the theoretical background, explained in detail. Many possible solutions did not have sufficient theoretical background or were not explained clearly.
- Simplicity: Most of the solutions are relatively simple, especially the Excel spreadsheets. If the software is well-documented, that also makes the solution simpler.
- MAV-Scale Implementation: Some of the solutions were used in the academic world to analyze UASs and MAVs. Some have real-world applications.
- Modifiability: Html-based tools, commercial software and some other freeware cannot be modified. Sometimes, it is required to modify input files and maybe subroutines as well.

- Tool Functions: Some tools have only a single function whereas some of them are very sophisticated and have many functions.

Based on those criteria, DATCOM (aerodynamics, stability and control), QPROP (propulsion) and AVL (stability and control, aerodynamics) were chosen. They are all well-documented and have sufficient theoretical background. The integratibility problem was solved using batch files.

3.1 Major Component Descriptions

This section provides the descriptions of the programs and detailed information on the components that have been used in this effort.

3.1.1 ModelCenter:"The Executive Control Program". Phoenix Integration's ModelCenter (MC) was designed as a model integration environment with its companion application "analysis server" [11]. MC provides a flexible framework to create an integrated design model and perform design optimization. Models and applications from outside environments can be brought together into MC, and it provides an intuitive graphical user interface (analysis view) to assist in building larger system models. Trade-study tools (2D or carpet plots) graphically display the data for the model that is being examined. In addition, MC has a wide variety of plug-in components such as Catia, Converger, Darwin, Design Explorer, Flames Analyze, Flames Execute, Gradient Optimizer, MATLAB, MathCad, NX, Optimizer, Excel, ProE, QuickWrap and Script. Importing outside applications can be achieved by direct interaction with MC or through analysis server connections [11] via FileWrappers, ScriptWrappers and ExcelWrappers as seen in Figure 3.2. Importing applications and applying plug-in features appropriately into the MCDA are the primary focus of this multidisciplinary research.

In addition to those primary functions, there are some other add-on packages available so as to provide additional optimization capabilities or existing tools can be

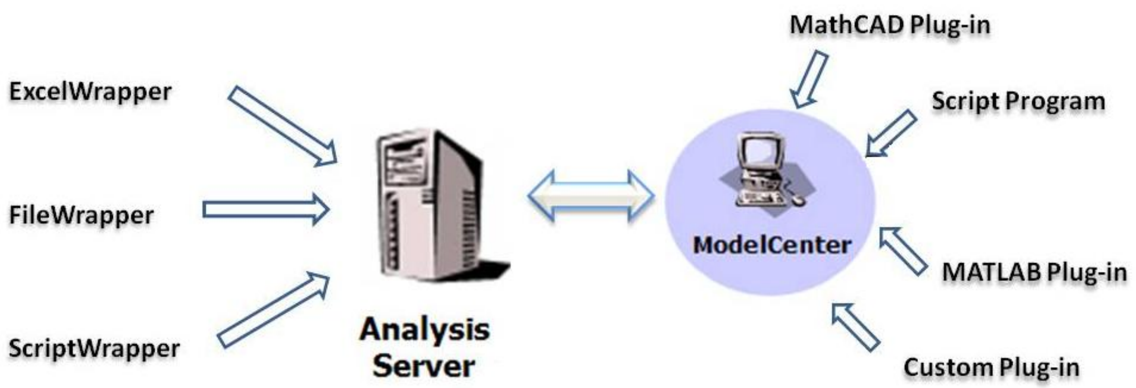


Figure 3.2 Importing Applications into MC

wrapped into the MC environment (for more applications, see reference [50]). Here is a list MC functions that were used in the development of the MCDA tool.

Outside applications:

- DATCOM Filewrapper
- QPROP Filewrapper
- AVL Filewrapper
- eHeli Excelwrapper

Plug-in applications:

- Aircraft Geometry Custom Plug-in
- MATLAB Plug-in (for various applications)
- Script Plug-in (for various applications)

Figure 3.3 summarizes the program environment and a short review of MCDA operation in MC is as follows: Aircraft components are created in the Aircraft Geometry (nose, body, aftbody, wing, horizontal wing, vertical wing, propellers) plug-in.

For aerodynamics, DATCOM filewrappers (with a selectively running script component) and MATLAB experimental data interpolation component; for propulsion, QPROP filewrapper; for stability and control, AVL filewrapper; and for the power requirement, eHeli Excelwrapper were created. There are a couple of MATLAB plug-ins and script files that create proper inputs and outputs for those major components i.e. filling the gaps between them. The geometry view not only provided an intuitive picture of the model but also it helped to verify the geometry was being built properly. Using MC link editor, variables are linked in an orderly fashion. Trade studies and modification based on the user defined inputs were made after creating the entire model.

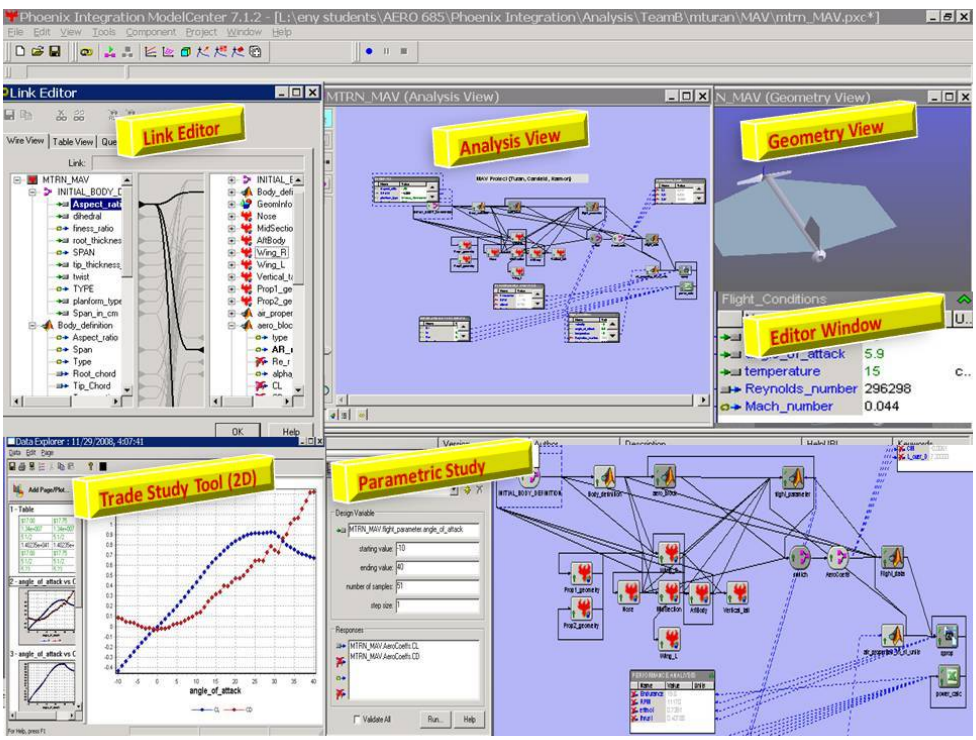


Figure 3.3 MC Work Space Overview

Some examples of the inputs are b, AR, planform type (Zimmerman, inverse Zimmerman, Rectangular, Elliptical), flight velocity (u), AoA, temperature (T_{celc}), QPROP inputs such as battery capacity, maximum load voltage, speed constant (K_v), zero torque current (I_o) and resistance (R) for the motor. In fact, it is up to

the user to select which variables are going to be inputs or outputs, because some of the components allow the user to vary the inputs such as QPROP. Once every link has been established, the user can trade-study any of those inputs mentioned versus any output variable. Some examples are C_L vs AoA, L vs T_{celc}, R vs endurance, etc. As seen, it is a very flexible environment with a lot of outputs for the user. Switching between analysis view and geometry view helps the user see the effects of the desired input visually. Also, carpet plots can be generated with two inputs and any reasonable output. Since everything is linked together, MC runs the trade study by pre-validating every single component which is linked to variables of interest which makes it a very powerful tool.

3.1.2 Digital DATCOM. Digital DATCOM was used in primary design operations for rapid and economical estimations of aerodynamic stability and control characteristics that are frequently required. The fundamental purpose of the USAF stability and control DATCOM is to provide methods for estimating stability and control characteristics in preliminary design applications. It calculates static longitudinal and lateral stability, dynamic derivatives and high lift and control surface characteristics. Trim option is also available [39]. Basically, it allows the user to estimate the design aerodynamic coefficients of an aircraft either from a design or for an existing aircraft. A user defined input file is executed via digdat.exe then an output file is created (Figure 3.4). See Appendix D.9 for examples of input, output and filewrapper code and see References [39, 13] for more information about DATCOM.

There is a derivative of DATCOM called DATCOM+. It is available online, with a front-end and back-end added to the original DATCOM for user convenience. By adding a different format output section to the original program, the data is output in various formats [2]:

- Free-format LFI tables, for plotting with LFIPLOT.
- XML format, compatible with JSBSim

Figure 3.4 Digital DATCOM Operations

- AC3D Model

But there are some known issues with DATCOM+. One is that defining airfoils manually (with upper and lower surface points), rather than using NACA numbers, does not provide any output for the AC3D picture. The second one is that fuselages are not drawn correctly if defined as other than a circular cross-section. Even with these restrictions, it is still user-friendly and produces the same data. In the MCDA tool, the original Digital DATCOM executable file is used. Using the plus (+) version did not give any additional capabilities due to the user defined airfoil (flatplate).

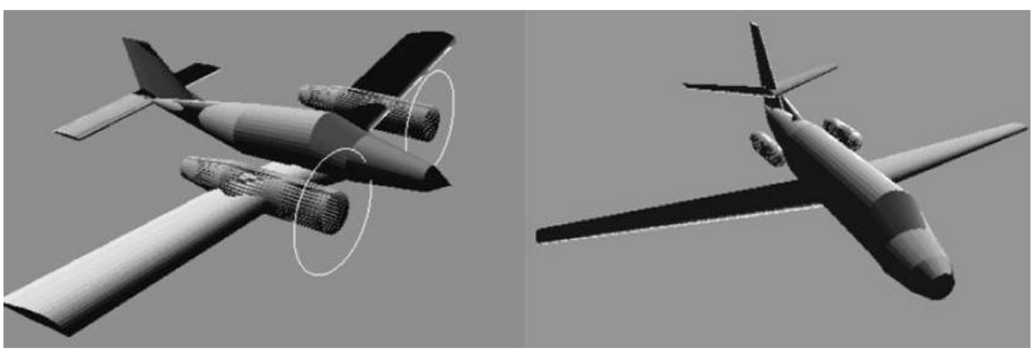


Figure 3.5 Examples from DATCOM+ AC3D View [2]

DATCOM requires certain parameters as inputs that go into the calculations. For example, DATCOM handles straight-tapered, cranked or double-delta wings. For Zimmerman, Inverse Zimmerman and elliptical planforms, it is not possible to

enter planform properties due to the limitation on inputs. Measures were taken to alleviate this problem and geometry inputs were created using some assumptions.

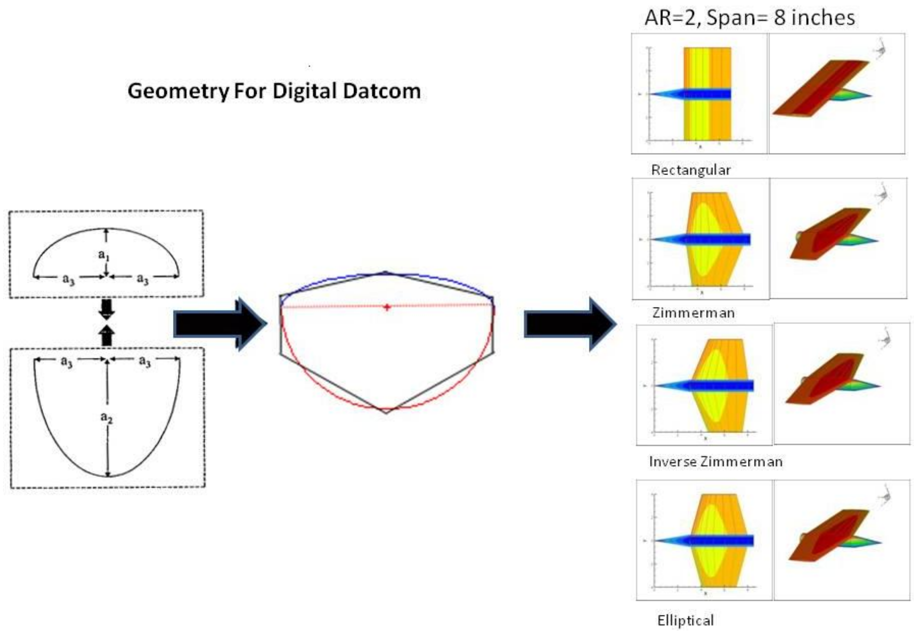


Figure 3.6 DATCOM Input Results

The first assumption is that an equivalent trapezoidal wing planform based on AR and b is sufficient and the second one is that the equivalent body based on fineness ratio for a body can be used. Zimmerman planform geometries are generated by joining two half-ellipses at the quarter-root-chord location. As plotted in Figure 3.6, one ellipse has semi-major axis a_3 and semi-minor axis a_1 while the other has semi-major axis a_2 and semi-minor axis a_3. Based on that calculation, a simple area rule was applied to Zimmerman, ellipse planform geometry (i.e. equivalent trapezoidal are for both of upper and lower half ellipses). A MATLAB m-file was generated for that purpose and inserted into MC for the geometry analysis part of the model. The results are tested in the DATCOM program and it was possible to observe the output of the DATCOM input file by using *Digdat Plot* 2007 Plotter. Figure 3.6 shows the planforms that were created for DATCOM via previously mentioned assumptions. DATCOM outputs are also presented in the right hand side of the same figure in

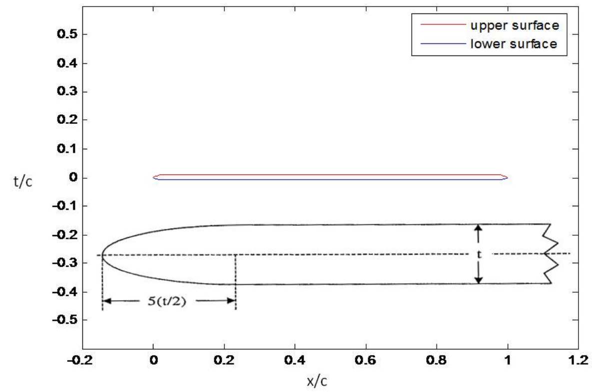


Figure 3.7 Flat Plate Profile

color. For the cases in this figure, Zimmerman, inverse Zimmerman and ellipse have a taper ratio of 0.57 being close to a taper ratio of 0.45 which almost completely eliminates the undesired effects for an unswept wing and produces a lift distribution very close to the elliptical ideal [54] in classic aerodynamic applications.

There are three options to define an airfoil section in DATCOM: an airfoil section designation (For NACA, double wedge, circular arc or hexagonal airfoils), section upper and lower Cartesian coordinates or section mean line and thickness distribution. Since the airfoil section in Torres research [59] was not defined with a designation, section upper and lower Cartesian coordinates were created in MATLAB and applied to DATCOM (Figure 3.7). All planform types in that research had leading and trailing elliptical edges of 5-to-1 and thickness-to-chord ratios (t/c) of 1.96%. For the body shape, it is cylindrical based on the fineness ratio, but there is an option to enter user defined body shapes manually in DATCOM too.

Also, the Digital DATCOM manual [39] warns of poor accuracy below Re$\cong 10^5$. Parametric studies in the MCDA tool revealed that DATCOM would generate the same aerodynamic data below Re$\cong 10^5$ and should not be trusted. Another test case was run at very low speed for a rectangular planform with 8 in span, AR=2

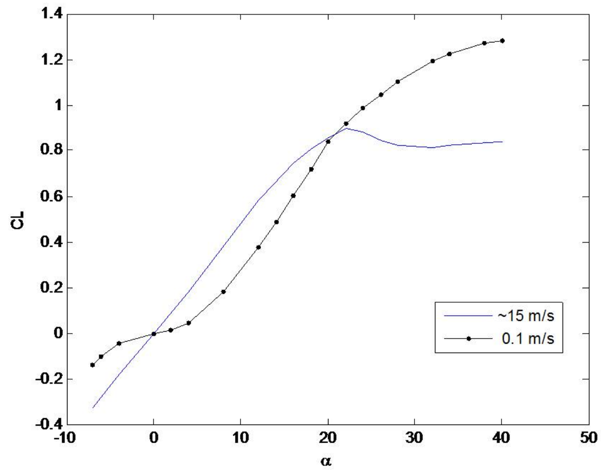
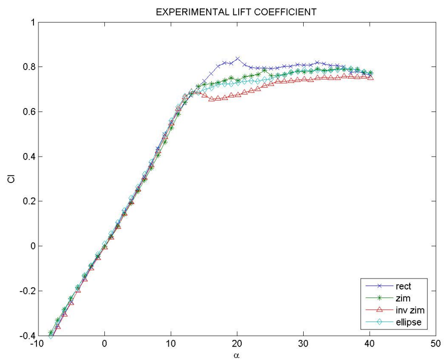


Figure 3.8 DATCOM Low Speed Test Run

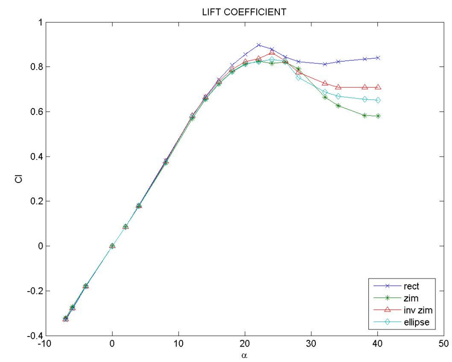
and 4 inches of chord at 15 and 0.1 m/s and DATCOM failed to operate at 0.1 m/s and only produced C_L data (Figure 3.8).

Digital DATCOM is not capable of calculating aerodynamic coefficients at very low Reynolds numbers, but there is an option to input experimental data such as C_L, C_D and C_M for the wing, body or combination of both. A DATCOM test case was run at Re=100K, AR=2 for all planforms to compare the results with experimental data. As shown in Figure 3.9, C_L and C_D curves had similar pre-stall tendencies except for the rectangular planform C_D. C_M and post-stall results did not seem to match at all because the DATCOM program was not designed to handle LAR wings at very low Reynolds numbers.

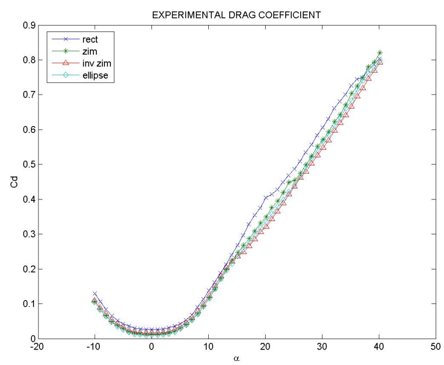
As a result, in order to utilize DATCOM's synthesizing capabilities, experimental inputs (interpolated) were used within the database limits and extrapolated up to a Reynolds number at which the result did not deteriorate very much. Beyond that Reynolds number, options are to use DATCOM stand alone or to supplement the database with additional experimental inputs. For example, C_D and C_M experimental input data (extrapolated) are used and the C_L is generated by DATCOM.



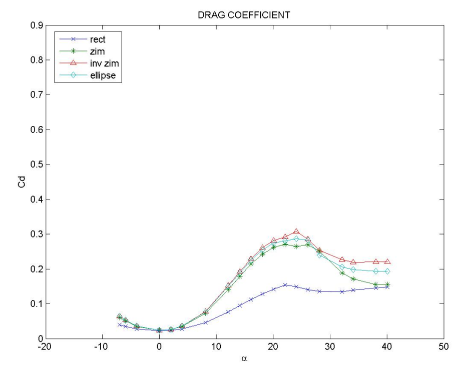
(a) Expr.C_L



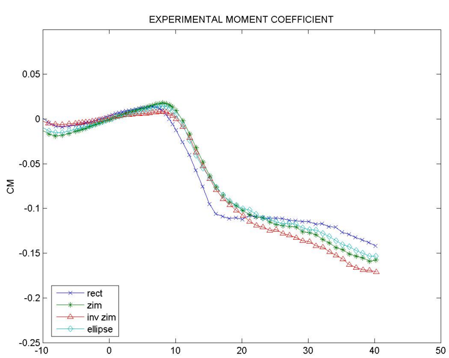
(b) DATCOM C_L



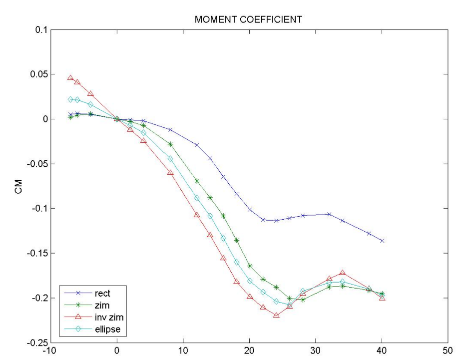
(c) Expr.C_D



(d) DATCOM C_D



(e) Expr.C_M



(f) DATCOM C_M

Figure 3.9 Comparison of Experimental Results [47] and DATCOM

3.1.3 QPROP. MAV propulsion systems must be as lightweight and efficient as possible. Small-scale propulsion systems will have to satisfy extraordinary requirements for high energy density and high power density depending on the mission. As long as there is no hovering requirement, fixed-wing propeller-driven MAVs have been found to be the most energy efficient [46]. Technologies like MEMS, low power electronics and component multi-functionality will help the performance of propeller-driven vehicles [40]. The R/C community now has many examples of light weight-hover capable or propeller-driven small indoor and outdoor vehicles.

In this research, battery driven propulsion systems for MAV prototypes are examined and QPROP is integrated into MC. In general, different outputs can be extracted, but those values are dependent on the selected motor, propeller and the flight conditions, based on the inputs such as required thrust (T_{req}) and u (Figure 3.10). All of those different database files can be manipulated easily within the MCDA using a fileWrapper component.

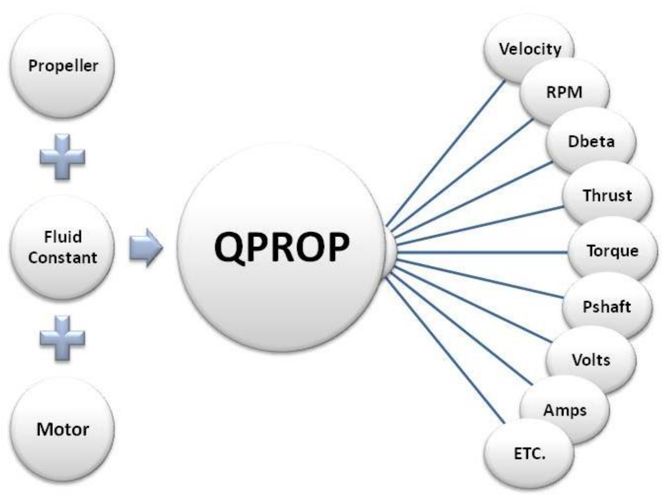


Figure 3.10 QPROP Operation

Drela's QPROP predicts the performance of propeller-motor combinations and assumes a brushed DC motor type, therefore it is limited. It also takes atmospheric conditions into account. Most of the commercial programs use relatively simple

propeller and motor models. QPROP has a relatively sophisticated and accurate propeller aerodynamic model and a general motor model. QPROP has two propeller/motor file formats: simple and advanced propeller/motor input files. Another good feature of QPROP is that the user can create his own input data for the propeller and motor. Although it has limited motor types, any motor model can be coded in SUBROUTINE MOTORQ (in motor.f). Moreover for non-electric motors, the voltage (V), passed to MOTORQ, can represent any suitable power-control variable, e.g. throttle setting, fuel flow rate, etc. [17, 18].

QPROP can be incorporated into MC via a filewrapper and can generate a wide variety of outputs as shown in Figure 3.10. The user guide explains the propeller aerodynamic model, QPROP theory, motor models and relations between equations, measurements for sophisticated propeller geometry and blade airfoil. There are three input files: fluid constant file, propeller file and motor file. Execution is performed via a batch file which is also considered as an input file in the MCDA. QPROP can be run in many different modes via single-point or multi-point runs. It was decided that the MCDA tool use a single-point run execution instead of a multi-point run. The reason for that is to keep the process simple and also, the MC parametric study tool can be used to generate data as if it were a multi-point run. The MCDA single-point run is based on the variables u and T_{req}. Finally, the results can be viewed on the screen or dumped into a text file. Another single-point run QPORP filewrapper was generated to evaluate QPROP in the parametric study tool in the coaxial MCDA and it allows the user to operate QPROP with its all functions once input files are supplied. See Appendix D.8 for examples of inputs, output and the QPROP filewrapper.

3.1.4 AVL. AVL software is a vortex lattice code developed by Mark Drela and Harold Youngren at the Massachusetts Institute of Technology (MIT). Vortex Lattice Methods (VLM) like AVL performs reasonably well for aerodynamic configurations which consist mainly of thin lifting surfaces at small AoA and side

slips (β) [19]. It is usually used in simulator modeling and in general it has been found useful for modeling unusual aircraft configurations [12] as seen in Figure 3.11.

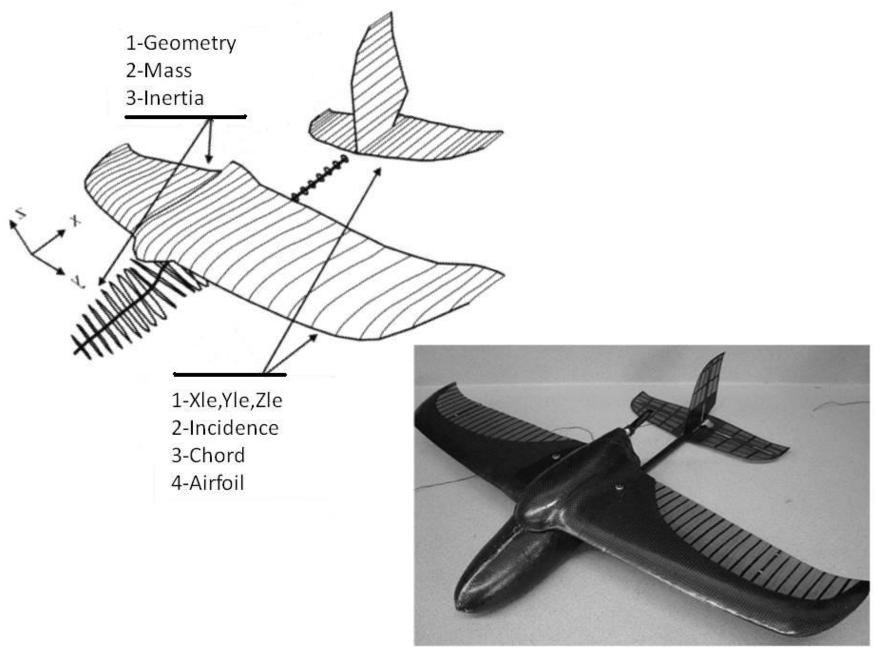


Figure 3.11 AVL Geometry Input Example, UF MAV [57]

AVL needs three input files: geometry, mass and run-cases. Unusual geometries can be defined in xxx.avl file and mass file (xxx.mass) requires a detailed analysis of components for vehicle of interest. Run-case (xxx.run) file allows user to evaluate the vehicle at various attitudes. As summarized in Figure 3.12, AVL generates the following data: run case aerodynamic coefficients (Cl_{tot}, Cd_{tot}, Cd_{ind}, Cl_{ff}, Cd_{ff}), control surface deflections (for flap, aileron, elevator and rudder) and stability axis derivatives (for wing, flap, aileron, rudder and elevator). See Appendix D.9 for examples of inputs, output and the AVL filewrapper.

The geometry of an aircraft is specified as the locations of each lifting surface of the aircraft including control surfaces. Moreover, airfoils can be created for those surfaces by Drela's XFOIL. Just for this purpose, Cloudcap Tech [12] company created an AVL Editor application which allows user to create an AVL model using a

GUI which can also call both XFoil and AVL. The geometry in AVL can be graphically represented in 3D, similarly to MC and the user can evaluate his/her inputs by looking at the geometry view. The stability derivatives about the center of gravity (CG) are calculated using the lifting surface geometry. The AoA or lift coefficient of the aircraft can be varied for different flight conditions then the stability derivatives are determined for each angular position [30]. See Reference [19] for detailed information and Reference [12, 57, 30] for various applications.

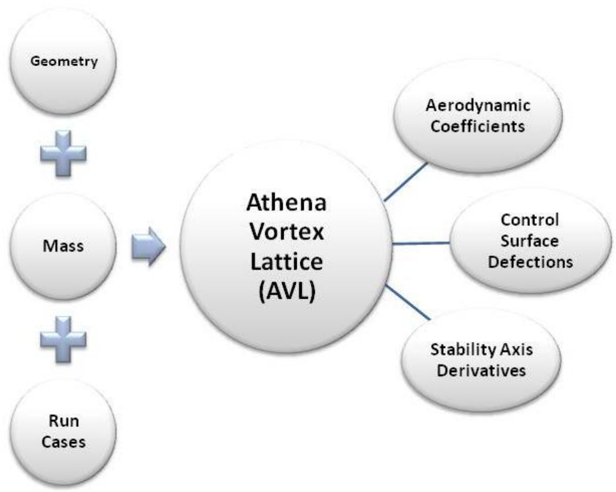


Figure 3.12 AVL Operation

3.1.5 MATLAB/Excel. MATLAB (MATrix LABoratory) is a tool to do numerical computations, display information graphically in 2D and 3D, and solve many other problems in engineering and science. Some of the features of MATLAB are: easy matrix manipulation, implementation of algorithms, implementation of algorithms, plotting of functions and data, etc. Microsoft Excel consists of a proprietary spreadsheet-application written and distributed by Microsoft and it has calculation tools, graphing tools, pivot tables, and other features.

These tools have a great variety of applications that were incorporated into the MC environment. Some examples are: experimental databases are in an Excel spreadsheet, a MATLAB m-file was created to obtain numbers (data) from that

spreadsheet and after that another MATLAB m-file created a 3D database and produced aerodynamic coefficients based on user-defined AR, Reynolds number and AoA by interpolating and extrapolating. Another example is eHeli, an Excel spreadsheet that was wrapped within MC.

For the author, these two tools have high importance. The reason for that is MC handles Excel and MATLAB plug-ins very well. Any related research about MAVs accomplished in MATLAB and Excel can be imported into the MC environment with slight modifications. It is up to the researcher's imagination which variables or equations are going to be integrated into the tool. Eventually this will help the integration process become very adaptive and flexible.

3.2 Major Component FileWrapper Structures

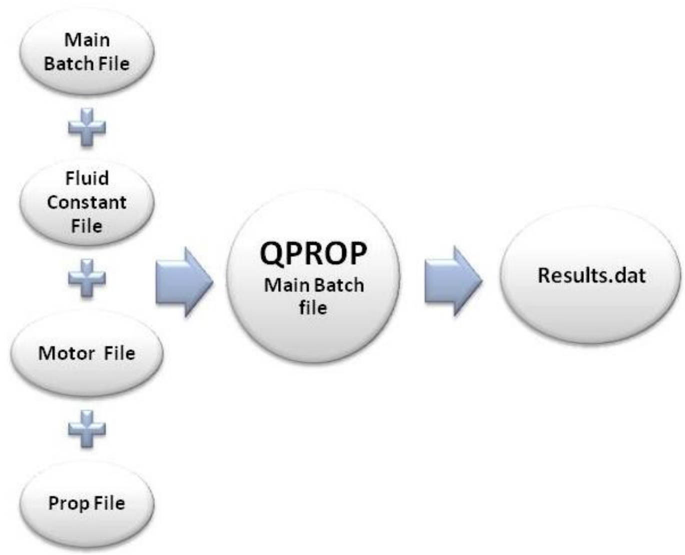
It is important to mention that the most significant challenges during the development of MCDA were the filewrapper structures of DATCOM, QPROP and AVL. The logic behind a filewrapper in MC is simple but manipulation of the input files, constructing the executables via batch file and creating an output file made the process very complicated. The logic behind the filewrappers and how they work is as follows:

> The FileWrapper utility enables users to create Analysis Server components from file I/O programs. These components are often referred to as FileWrapper components. A file I/O program is an analysis that has one input file, an executable that can be run from the command line of an operating system shell and one output file as in DATCOM example. More complicated file I/O programs may have multiple input files and/or output files (as in QPROP and AVL) associated with either a single executable or multiple executables. The FileWrapper utility is designed to automate the execution of analyses that are based on file I/O programs. A user can create a FileWrapper component that will automatically edit the appropriate input file(s), run the executable(s), and parse the output file(s) of an analysis whenever the component is executed by the Analysis Server [50].

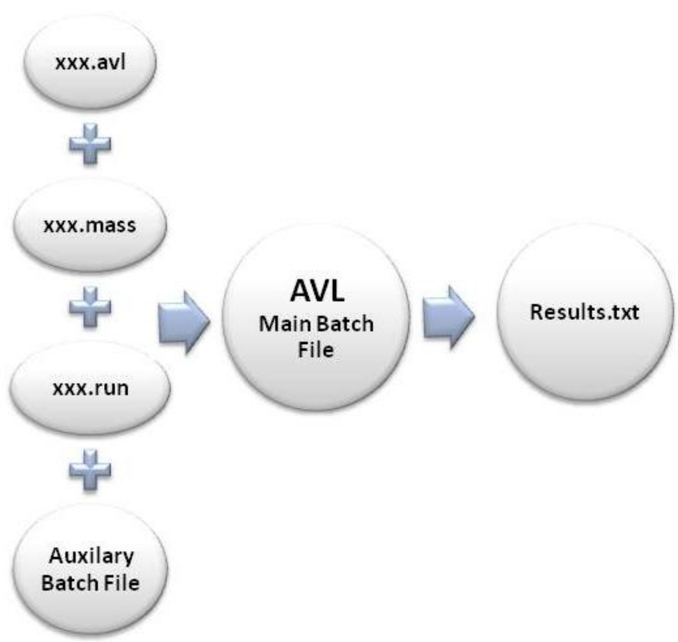
Figure 3.13 summarizes the filewrapper structures of DATCOM, QPROP and AVL in general. As explained in the previous paragraph, the fileWrapper component will automatically edit the input file(s) on the left hand side of the figures, run the executable or batch file in the middle and parse the results as seen on the right hand side of the figures. As seen in Figure 3.13, DATCOM is the easiest to handle and AVL is the hardest one in terms of inputs and outputs.

The latest version (v.8) of MC has a filewrapper plug-in called QuickWrap which helps user create filewrappers with a user-friendly interface. The QuickWrap is stored under the component plug-ins item of the Server Browser. Specifying the input/output files and selecting the variables in QuickWrap can be done via three ways: the auto-import tool, point-and-click specification and manual creation. See the related Appendices D.7.3, D.8.3 and D.9.3 for filewrapper templates.

(a) DATCOM Filewrapper Structure



(b) QPROP Filewrapper Structure



(c) AVL Filewrapper Structure

Figure 3.13 Filewrapper Structure Overview

3.3 Integration of The Tools

In this section, the pieces of the fixed wing MCDA tool are explained. The related code may be found in the corresponding Appendices. Components names are arbitrarily chosen by the author during the code development and can be renamed differently in future applications.

3.3.1 Geometric Properties.

This function is accomplished under two components which are the "Initial-Body Definition (MC component)" and the "Body Definition (MATLAB)" depicted under the the component tree. Geometric properties of the vehicle for the equivalent wing and body were primarily needed as inputs for DATCOM and aircraft geometry components, but used in several other components as well.

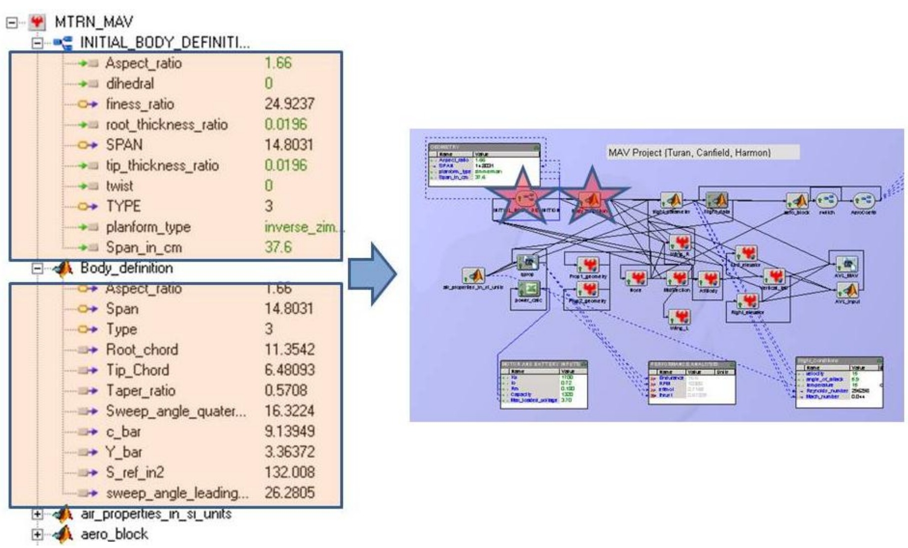


Figure 3.14 Geometric Properties Overview

The "Initial-Body Definition" is the component where a user defines AR, span (b) and planform type for the MCDA tool. There are also other inputs such as dihedral, twist, tip and root thickness ratio, but they do not have a direct effect on the MCDA analyses other than to change the geometry of the vehicle in geometry

view. These extra inputs are in the predefined Aircraft Components (wing, elevator, rudder, fuselage) and they are not incorporated into any analysis in the MCDA tool.

The second component is the "Body-Definition" m-file which calculates the geometric properties, when AR, b and type (Rectangular, Zimmerman, inverse Zimmerman and Elliptical) are defined. As summarized in Figure 3.15, the m-file will output root chord (c_{root}), tip chord (c_{tip}), mean aerodynamic chord ($\bar{c}$) and Y location of $\bar{c}$ ($\bar{Y}$), sweep angle (Λ) and taper ratio (λ) of the respective wing planform via a simple area rule based on the assumptions mentioned in Chapter 3.1.2. The equations derived from those assumptions are as follows:

$$S_{ref} = \frac{b^2}{AR} \tag{3.1}$$

$$a_3 = \frac{b}{2} \tag{3.2}$$

$$a_1 = 3 * a_2 \tag{3.3}$$

$$c_{root} = a_1 + a_2 \tag{3.4}$$

$$c_{tip} = (a_1 + a_2) * (\frac{\pi}{2} - 1) \tag{3.5}$$

$$c_{tip_{upper}} = a_1 * (\frac{\pi}{2} - 1) \tag{3.6}$$

$$\bar{c} = \frac{2}{3} * c_{root} * \frac{1 + \lambda + \lambda^2}{1 + \lambda} \tag{3.7}$$

$$\bar{Y} = \frac{b}{6} * \frac{1 + 2 * \lambda}{1 + \lambda} \tag{3.8}$$

$$\Lambda = \arctan\left[\frac{a_1 - c_{tip_{upper}}}{a_3}\right] \tag{3.9}$$

$$\lambda = \frac{c_{tip}}{c_{root}} \tag{3.10}$$

Rectangular and ellipse planform parameters are inherently easy to find due to symmetry. The (inverse) Zimmerman planform geometries are generated by joining

two half-ellipses at the quarter-root-chord location. One ellipse has semi-major axis a_3 and semi-minor axis a_1 while the other has semi-major axis a_2 and semi-minor axis a_3.

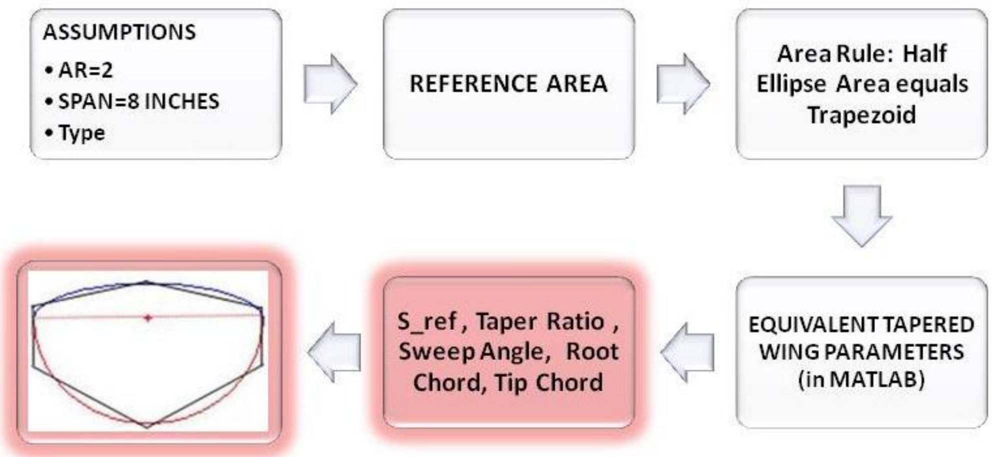


Figure 3.15 Geometry Calculation Overview

3.3.2 Atmospheric Model. This component is named as "Air Properties in SI Units". The Atmospheric Model Component is an intermediate component which will calculate the air properties at operational conditions when user defines the ambient temperature (T_{celc}).

There is only one direct user input which is T_{celc}. Then temperature in Kelvin (T_{kel}), pressure (P) and speed of sound (a) are calculated via the following equations [8]. (R= 287 and γ= 1.4)

$$T_{kel} = T_{celc} + 273.15 \tag{3.11}$$

$$P = \rho * R * T_{kel} \tag{3.12}$$

$$a = \sqrt{(}\gamma * R * T_{kel}) \tag{3.13}$$

The variables density (ρ), kinematic viscosity (ν) and dynamic viscosity (μ) are interpolated in a MATLAB plug-in based on the database in Figure 3.16 of air properties in SI units.

Temperature T, $[^\circ C]$	Density ρ, $[kg/m^3]$	Dynamic Viscosity $\mu \times 10^5$, $[N \cdot sec/m^2]$	Kinematic Viscosity $\nu \times 10^5$, $[m^2/sec]$
0	1.29	1.72	1.33
5	1.27	1.74	1.37
10	1.25	1.77	1.41
15	1.23	1.79	1.46
20	1.21	1.81	1.50
25	1.19	1.84	1.55
30	1.17	1.86	1.60
35	1.15	1.88	1.64
40	1.13	1.91	1.69
45	1.11	1.93	1.74
50	1.09	1.95	1.79
55	1.08	1.98	1.84
60	1.06	2.00	1.88
65	1.04	2.02	1.93
70	1.03	2.04	1.98
75	1.01	2.07	2.04
80	1.00	2.09	2.09
85	0.987	2.11	2.14
90	0.973	2.13	2.19
95	0.960	2.15	2.24
100	0.947	2.17	2.30

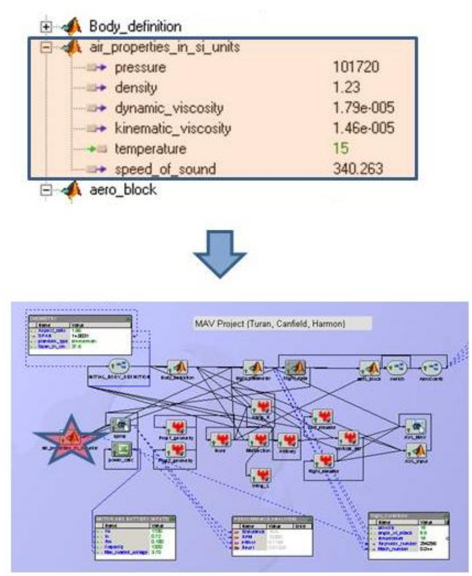


Figure 3.16 Properties of Air [6] and Atmospheric Model Component Overview

Consequently, those outputs are used as flight condition inputs (in the QPROP), the calculation of forces (in the Flight Data Component) and finding M, a, Re number (in the Flight Parameter Component).

3.3.3 Experimental Aerodynamic Data Interpolation/ Extrapolation. This function is accomplished under the "Aero-Block" component. The idea behind the Aero-Block was that DATCOM would not produce good results at very low Reynolds numbers as specified in the manual. However, there is an option in DATCOM that the user can input their experimental data in order to override the DATCOM results. Extensive experimental data was needed for that purpose. Figure 3.17 summarizes the Aero-Block component.

The aerodynamic data on various airfoil geometry and wing planform of the lifting surfaces are very important. Due to the lack of these type of data, phys-

ical or numerical experiments are needed. Drag calculation is more difficult due to the order of magnitude being smaller than the lift. LAR wing theory and experimental data by Mueller and Torres have been used to analytically predict the performance of the MAV [46]. For very low Reynolds number aerodynamics, Mueller and Torres [47] conducted experiments on LAR wings and they collected data for rectangular, Zimmerman, inverse Zimmerman and elliptical planforms, ARs of 0.50, 0.75, 1.00, 1.25, 1.50, 1.75 and 2.00 and Reynolds numbers of 70K, 100K and 140K. Details of the experimental setup and how it was conducted can be explored in the related references [47, 59]. Those experimental data was converted into a more functional format and processed in order to be utilized in the MCDA tool.

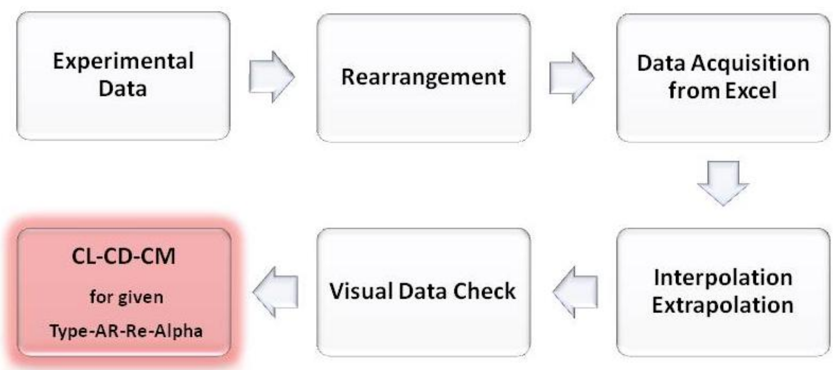


Figure 3.17 Aero-Block Overview

Steps are as follows:

- **Experimental Data [47] for DATCOM EXPR input**

  The experimental data from Torres and Mueller is provided in an Excel file and had C_L, C_D and L/D tabulated as a function of α for the rectangular, Zimmerman, inverse Zimmerman, and elliptical planforms of ARs 0.50, 0.75, 1.00, 1.25, 1.50, 1.75, and 2.00. The Reynolds numbers available were 70K and 100K based on c_{root}. Also the 95% uncertainty bounds were listed for C_L, C_D and L/D which were labeled as dC_L, dC_D and dL/D, respectively.

The second Excel spreadsheet had C_N and C_M in a similar fashion except that the Reynolds numbers were 100K and 140K.

- **Rearrangement of Experimental Data**

  Rearrangement of the data was needed for acquiring them automatically via MATLAB functions. In order to do that, $C_L,\ C_D\ and\ C_M$ data were extracted from related columns and three separate Excel spreadsheet were created under each aerodynamic coefficient name (data tabulated in a similar way mentioned above). There was one additional change to the C_L Excel spreadsheet which is C_L data at Reynolds number 140K. Data for C_L at 140K was calculated from normal force coefficient (C_N) data vs α at 140K [9]. The same procedure was not applied to the C_D data because there was an unknown axial force component associated with wind the tunnel data [9].

- **Data Acquisition from Excel files**

  This step was accomplished by using the "xlsread" function in MATLAB and saved as "xxx.mat" for future data callings.

- **Interpolation/Extrapolation of the Experimental Data**

  The planform names were given a type number (Type 1 Rect - Type 2 Zim - Type 3 Inv Zim - Type 4 Ellip). A MATLAB m-file was generated for interpolation and extrapolation based on the type of planform, AR, Reynolds number and α. An example of the interpolation function in MATLAB used for C_L after calling the related planform Cl.mat file is:

  CLi=INTERP3($AR_X,\ alpha_Y,\ Re_Z,\ CL,\ AR_r,\ alpha_r,\ Re_r$, 'spline')

- **Visual Data Check**

  It was important to check if the database was created properly and Figure 3.18 has examples of a 4D plot of the experimental $C_L, C_D\ and\ C_M$ for Zimmerman planform. Plots were created by Jayaraman's m-file [31] in order to check the

meshing process results visually. Customizable four-dimensional plots were created using MATLAB's function "slice" by Jayaraman.

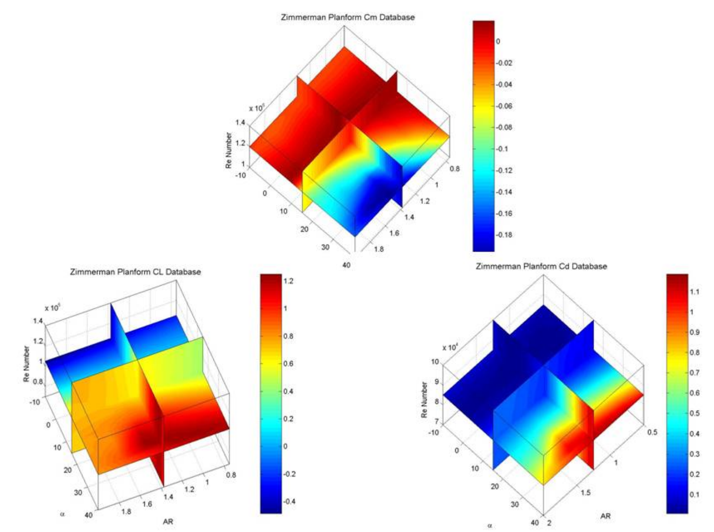


Figure 3.18 Interpolated C_L, C_D, C_M data for Zimmerman Planform

Now C_L, C_D *and* C_M can be extracted for any Reynolds number, AR and α within the data limits. Extrapolating also can be done but the results seems to vary a lot. Those interpolated data then were used as experimental inputs for DATCOM at low Reynolds numbers. Procedures that were followed for aerodynamics are summarized in the MCDA tool in Figure 3.19.

3.3.4 Determination of Aerodynamic Coefficients. This function is accomplished by one of two components which are selectively run according to an MC "Switch" and "Aero-Coefficients" components. The logic behind selectively running an MC script is that it will run one of the two different DATCOM filewrappers dependent upon the Reynolds number.

As seen in Figure 3.20, Selectively Running MC "Switch" component consists of three subcomponents which are: digdat_MAV filewrapper (with experimental C_L and C_D overriding), digdat_MAV_CL_only (with experimental C_D overriding) and

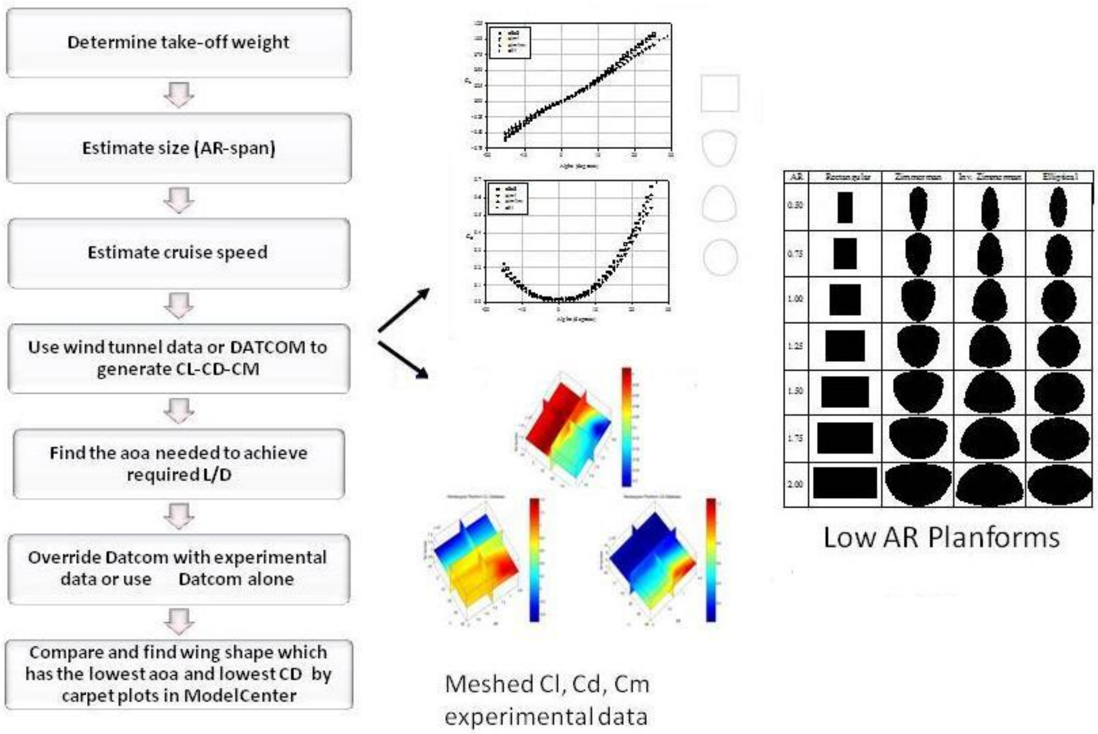


Figure 3.19 Steps followed for Aerodynamics

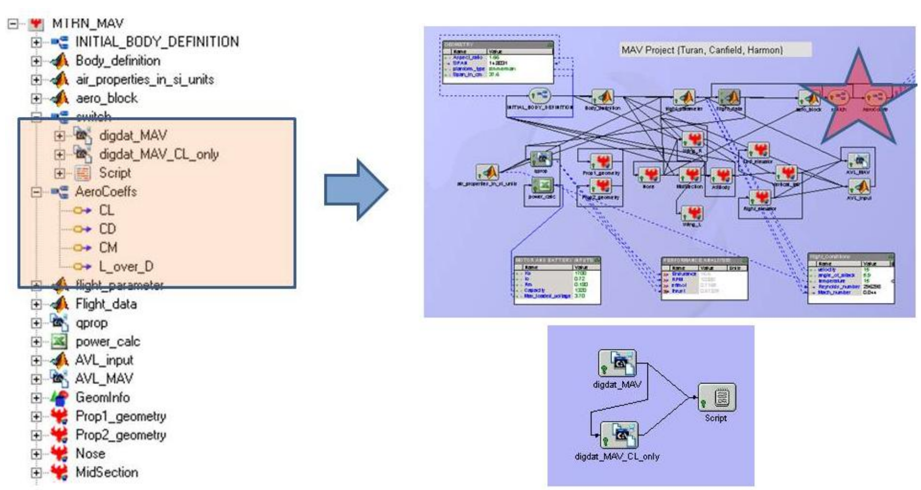


Figure 3.20 Selectively Running MC Switch and Aero-Coefficient Components Overview

a script file that determines which filewrapper to be used. Since the experimental data were available up to Re=140K and we want to extend the capabilities further, DATCOM results above a predetermined Reynolds number was used. Below that Reynolds number, Aero-Block generated experimental $C_L,\ C_D\ and\ C_M$ data would override DATCOM results. Some extrapolation cases were run in the parametric trade study tool in the development of Aero-Block Component and it was found that C_L data would deteriorate too much once a Reynolds number was picked well above the experimental data limit (Re=140K) but C_D and C_M would not be affected as much as in C_L case.

It is important to mention DATCOM operations. DATCOM is run via a filewrapper in the MC environment. There is only one input file and one output file. Although the user can define many variables for various calculations, DATCOM operation is kept simple and can be subject to change based on the user requests. It should be noted that filewrappers will only change the parameters that are specified in them but an actual DATCOM, AVL or QPROP input file(s) require several variables. So those variables other than those specified in filewrappers can be thought as "frozen variables". Although they are frozen, they have a direct effect on the results and user has to be cautious when configuring filewrappers and templates for filewrappers. The DATCOM filewrapper in the MCDA tool has the following variables:

- Flight conditions: M, Re, α
- Options: Reference area (S_{ref}), b
- Components: Wing Apex (X location), Wing Apex (Z location), CG (X location), CG (Z location)
- Fuselage: Fuselage cross section locations, Radius at cross section locations
- Wing: $c_{root},\ c_{tip},\ \lambda$, semi-span ($b/2$), exposed semi-span ($b/2$)
- Experimental data input: $C_L,\ C_D\ and\ C_M$ or $C_D\ and\ C_M$

- Results (Aerodynamic coefficients): $C_L,\ C_D\ and\ C_M$

It is possible to extend the current DATCOM filewrapper capabilities. Some of them are: dynamic derivatives for body, wing, wing body, wing-body-tail configurations; longitudinal trim data (for control device on wing or tail and all-movable horizontal stabilizer); power and ground effects; static and dynamic stability output (static longitudinal and lateral stability, dynamic derivatives). However, it is questionable if those applications can be used in MAV design or not due to the LAR wings at low Re numbers and propeller-induced flow characteristics. In the MCDA tool, only $C_L,\ C_D\ and\ C_M$ parameters are compared to experimental results. Therefore there are many options that can be discovered and accomplished in future research. At least, it is believed that some of the procedures can be used in MAV design. See reference [39] for DATCOM capabilities.

"Aero-Coefficients Component" is a companion component to the Switch component and it simply mirrors the $C_L,\ C_D\ C_M$ and L/D data in an organized fashion. These coefficients are used in lift (L), drag (D), and moment (M) calculations (Flight-data Component).

The Switch component might seem too complicated to the reader but this decision is given after taking too many constraints into account (assumptions, restrictions, limitations) for the project's progress. Aerodynamics was the most challenging part of this research. Priority was given to using experimental data which would have 3D effects.

Consequently, a tool that would calculate aero-coefficients is required for quick trade studies in future applications for the conceptual design of MAVs. The options are experimental data in a range that covers the entire flight regime of the MAV of interest, a AVL type-VLM code, a DATCOM for MAVs or some other codes such as XFOIL, XWING. Then those options would replace this complicated process.

3.3.5 Flight Parameters and DATCOM Input Converter. Flight Parameters Input Block is an intermediate component which will calculate the parameters related to the flight regime. Also there are two subcomponents which will prepare variables for DATCOM operations.

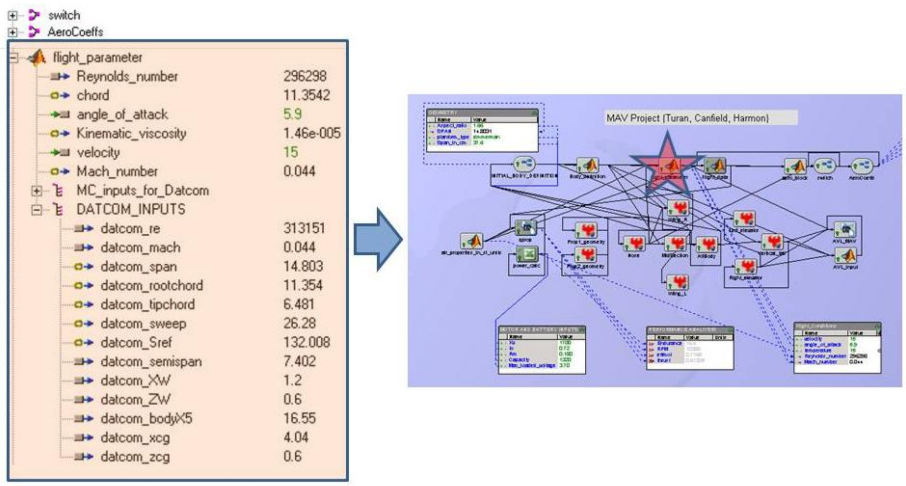


Figure 3.21 Flight Parameters Component Overview

As summarized in Figure 3.21, the direct user inputs are u, α and the indirect inputs are c_{root} (from right wing chord in inches), and ν (from air properties block in SI units). Then the Reynolds number and M are calculated within this component via the following equations:

$$Re = \frac{u * c_{root} * 0.0254}{\nu} \tag{3.14}$$

$$M = u/a \tag{3.15}$$

Also, there are two subcomponents named the "MC inputs for DATCOM" and the "DATCOM inputs" and their functions are as follows:

- **MC inputs for DATCOM:** The M, b, c_{root}, c_{tip}, Λ, S_{ref}, X_{cg}, Z_{cg} ,X_w and Z_w are variables for DATCOM, gathered in this subcomponent in order to

prepare variables for DATCOM operations. Refer to DATCOM manual [39] for input descriptions.

- **DATCOM inputs:** After manipulation, same variables are gathered in this subcomponent in order to keep track of the variables that will feed the DATCOM filewrappers.

In the MCDA tool, a user defines u and α under this component when running parametric trade study tool.

3.3.6 Flight Data Component. Flight Data component consists of two subcomponents which are the "Inputs for Forces" and the "Flight Data" results.

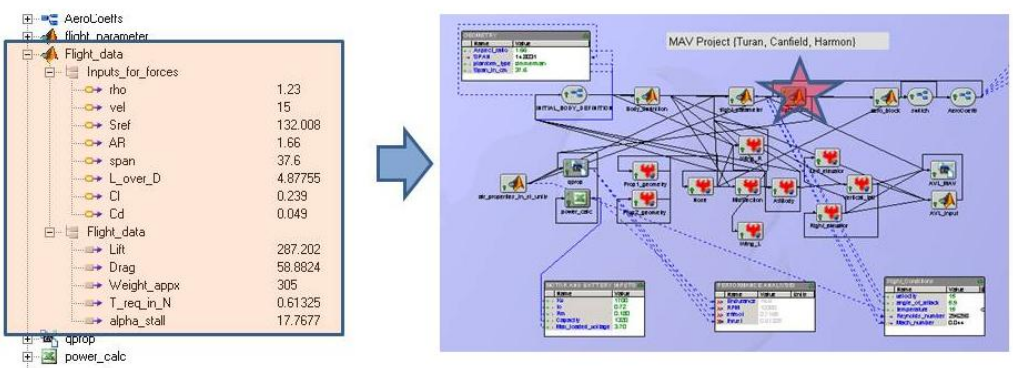


Figure 3.22 Flight Data Overview

The "Inputs for Forces" subcomponent gathers the relevant variables, i.e. ρ, u, S_{ref}, AR, b, C_L, C_D and L/D. Those variables are used in the calculation of L, D, approximate weight (W_{appx}), α_{stall}, and thrust (T) under the "Flight_data" subcomponent via the following equations after units were matched (g=9.807):

$$L = \frac{1}{2} * \rho * u^2 * CL * S_{ref} \tag{3.16}$$

$$D = \frac{1}{2} * \rho * u^2 * CD * S_{ref} \tag{3.17}$$

$$\alpha_{stall} = -10 * \arctan[4 * (AR - 1.25)] + 28 \text{ [47]} \tag{3.18}$$

$$\left(\frac{T}{W}\right)_{cruise} = \frac{1}{(L/D)_{cruise}} \text{ [54]} \tag{3.19}$$

$$T_{req} = \frac{W_{appx}}{(L/D)} \tag{3.20}$$

While using thrust matching, it is assumed that thrust is aligned with the flight path. In unaccelerated flight, the thrust must be equal to the drag; likewise, the weight must equal the lift [54]. In reality, there is a contribution of the dynamic thrust generated by the propeller to L and D.

3.3.7 QPROP. This component is the propulsion part of the the MCDA tool and it is highly sophisticated. The MCDA QPROP filewrapper is a simplified version that a user can tailor based on the different approaches to the propulsion part of the MCDA tool.

In the MCDA tool, QPROP will generate results based on the T_{req} and u. The u is a direct input by the user but T_{req} is calculated via Eq. 3.20. Then these inputs feed into QPROP and it will make its evaluation based on motor, propeller geometry and atmospheric conditions. These input files can be manipulated in MC environment as well. Finally, QPROP will generate the following data:

- Velocity $(u) = V$
- Propeller RPM (ω)
- Pitch Change in Degrees $(D\beta)$
- Propeller Thrust (T)
- Propeller Torque (Q)
- Shaft Power $(P_{shaft}) = Q * w \ and \ w = RPM * \pi/30$
- Motor Voltage (V)
- Motor Current (I)

- Motor Efficiency (η_{mot})
- Propeller Efficiency (η_{prop}) = $T * \frac{V}{P_{shaft}}$
- Advance Ratio (adv) = $\frac{V}{w*R}$
- Thrust Coefficient (C_T) = $\frac{T}{\frac{1}{2}*\rho*(w*R)^2*\pi*R^2}$
- Torque Coefficient (C_P) = $\frac{Q}{\frac{1}{2}*\rho*(w*R)^2*\pi*R^3}$
- Slipstream Velocity Increment (DV)
- Overall Drive Efficiency, eff (η) = $\eta_{mot} * \eta_{prop}$
- Electrical Power (P_{elec}) = amps*volts = $I*V$
- Propeller Power (P_{prop}) = $V*T$
- Power-Weighted Average Local (cl_{avg})
- Power-Weighted Average Local (cd_{avg})

Manually running the QPROP is a very tedious process and getting results requires some effort. This process is simplified in MC so that the user can easily run the cases.

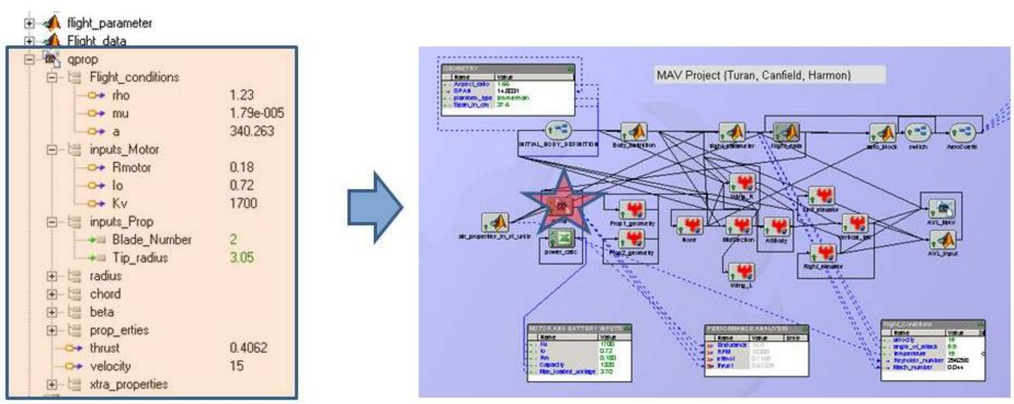


Figure 3.23 QPROP Overview

QPROP component consists of the following subcomponents as inputs:

- Flight Conditions: ρ, μ, a

- Motor Inputs: R, I_o, K_v
- Propeller Inputs: Blade number, tip radius
- Radius (r): for seven different cross sections
- Chords (c): for seven different cross sections at different radius
- Beta Angles (β): for seven different cross sections at different radius
- Velocity (u): Flight Velocity (i.e. incoming velocity to the propeller)
- Thrust (T_{req}): Required Thrust (i.e. thrust that propeller is supposed to generate)

  QPROP component consists of the following subcomponents as outputs:

- PROP_erties: $\omega,\ Q,\ P_{shaft},\ V,\ I,\ \eta_{mot},\ \eta_{prop}$
- Extra Properties: $adv,\ C_T,\ C_P,\ DV,\ \eta,\ P_{elec},\ P_{prop},\ Cl_{av},\ Cd_{avg}$

  Some of the relations used in the QPROP subroutine are:

$$Q = (I - I_0)/K_v \tag{3.21}$$

$$w = (V - I * R) * K_v \tag{3.22}$$

$$P = w * Q = (V - I * R) * (I - I_0) \tag{3.23}$$

$$eff = P/(I * V) = (1 - I * R/V) * (1 - I_0/I) \tag{3.24}$$

General discussion about QPROP is given is Section 3.1.3. QPROP is run via a filewrapper and is not a single-input/single-output type of tool. Flight conditions, motor, propeller and the batch file that runs the QPROP are all separate input files comprising many variables [17]. Although there are many data generation options in QPROP, the author restricted the QPROP Batch file run case only to T_{req} and u in the MCDA analysis.

3.3.8 Power Performance Calculator. This component calculates the endurance in minutes via an Excel spreadsheet called eHeli. It was originally designed for small rotary vehicles but adapted to the MCDA tool. The Current Cell in the eHeli spreadsheet is overridden with the QPROP current output. There are more complicated spreadsheets with databases available online like Power System Comparison [4] from the same source eHeli. Instead of using a spreadsheet like eHeli, a MATLAB m-file could have been used. But implementation of Excel spreadsheets might be more useful for future research in MC, since they are widely used in the R/C community.

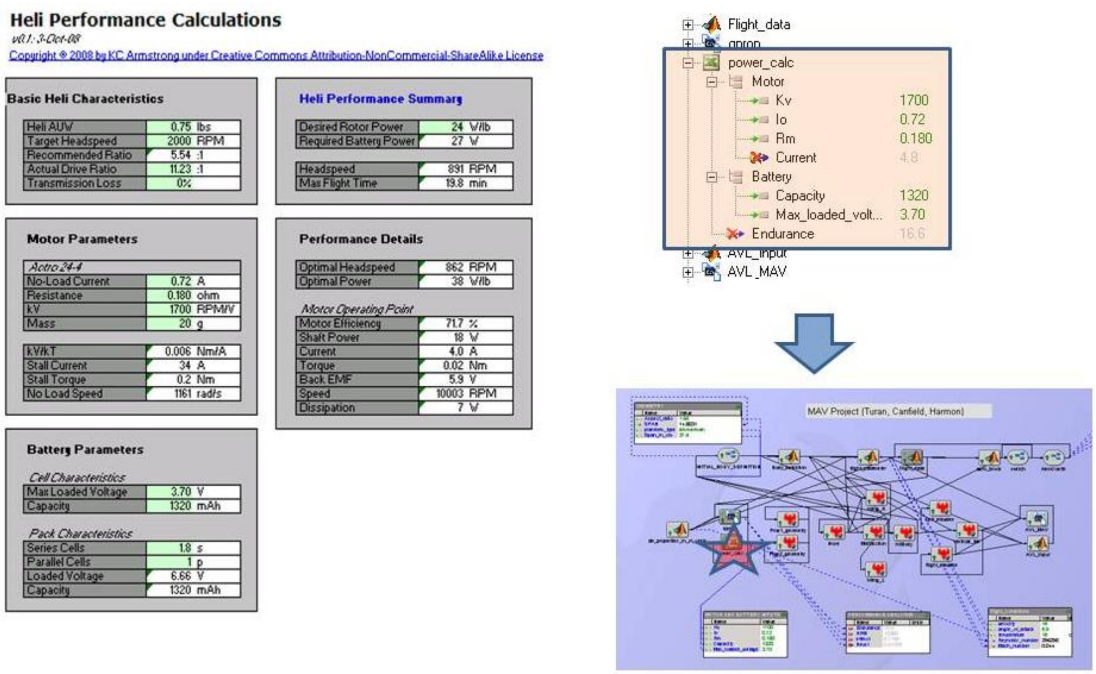


Figure 3.24 eHeli and Power Calculator Overview

The Power Performance calculator has two subcomponents which are the "motor" and the "battery properties". For the motor, the K_v, I_0 and R_m are input values; for the battery, capacity and max loaded voltage are the inputs. Motor properties also are fed into QPROP. Once QPROP runs and gives out the results, it feeds the power calculator with I_{req} (*current*). Then the power calculator uses motor, battery and I_{req} as inputs. Finally it will calculate the endurance in minutes.

3.3.9 AVL Operations. This component is used as the stability and control part of the MCDA tool. AVL operations are accomplished under two components; "AVL Inputs" and "AVL MAV".

The function of the "AVL Inputs" components is to gather and produce inputs for AVL. There are three subcomponents under the AVL Inputs named wing, horizontal stabilizer and vertical stabilizer. In the AVL manual, there is a caution for the modeling of bodies and it recommends leaving the body out of the AVL model if a fuselage is expected to have little influence on the aerodynamic loads. Therefore it was decided to leave the body out in the AVL component. Although many geometric variables from other components are linked to the "AVL Inputs" for the geometry generation, not all of them are defined. So missing variables for the geometry are calculated in this subcomponent. For instance, x, y, z locations for wing, h-stab and v-stab sections are calculated from sweep (Λ) and semi-span ($b/2$). Similarly, c_{tip} for h-stab and v-stab are calculated from taper ratio (λ) within this subcomponent.

"AVL MAV" is the core of stability and control analysis of the MCDA tool. It was the last piece added to the MCDA tool by the author and the most challenging one. This filewrapper has a unique structure and may serve as an example to other filewrapper operations. AVL has three input files: geometry (xxx.avl), mass (xxx.mass) and run-case save (xxx.run) input files as illustrated in Figure 3.12. Those input files are very detailed, only the geometry input file is mostly integrated into the MCDA. By saying mostly, the author means there are many other options that a user can specify for the geometry and this is true for the other input files in AVL. For example, creating the mass file itself a requires great effort by defining mass, x, y, z location, I_{xx}, I_{yy}, I_{zz}, I_{xy}, I_{xz} and I_{yz} for every single component on the vehicle i.e. nose, wings, rudder, wing connectors, battery, propeller, servos, rods, cables, pins, pods, autopilot and camera. But instead of typing those parameters manually, the MC environment can be used in future applications to generate these inputs as in the geometry (xxx.avl) input file example of the MCDA tool.

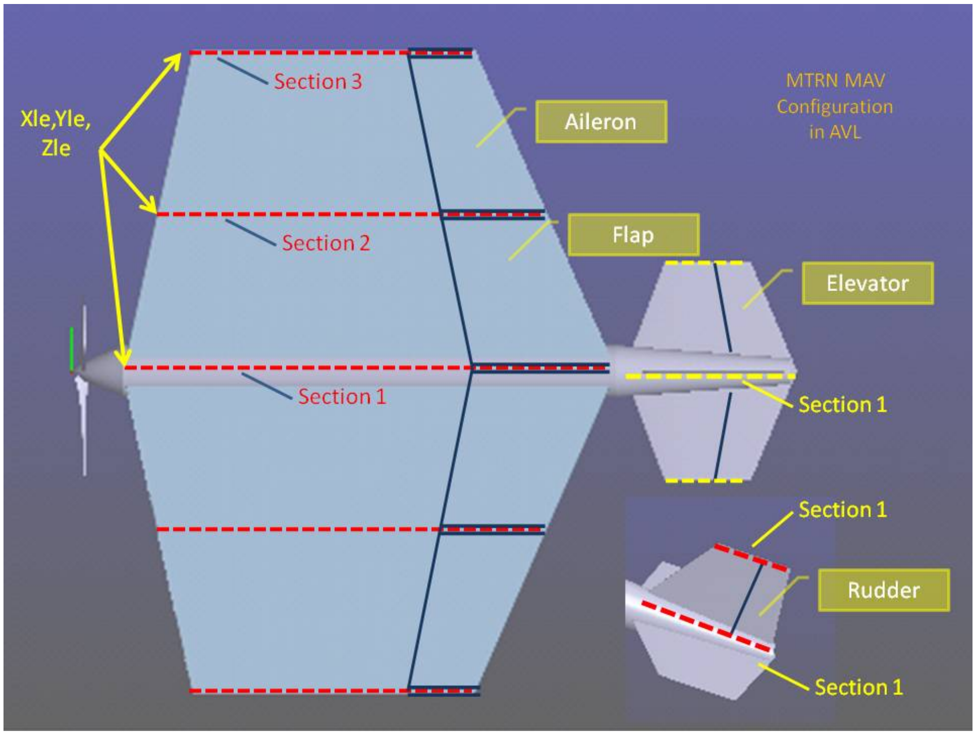


Figure 3.25 AVL Geometry defined in the MCDA Tool

Figure 3.25 illustrates the MCDA MAV geometry file (mtrn_MAV.avl) construction. For the wing, a partial-span control surface was specified by defining two panels, i.e. three sections, which have a flap over the inner panel and an aileron over the outer panel, whereas the rudder and elevator have only one panel, i.e. two sections. The CONTROL keyword in AVL geometry declares that a hinge deflection at this section is to be governed by one or more control variables. An arbitrary number of control variables can be used (but limited). Also non-symmetric control effects, such as Aileron Differential, can be specified in the geometry file. See *avl_doc.txt* file for a detailed description of input files [19].

AVL is normally executed from a command prompt but a batch file was created in order to integrate AVL into the MC environment. After the input files are processed, AVL will open up the main window where the user can pick different menus and under each command, there are sub-menus with many options. For instance, Oper menu has the following options:

• Select run case • List defined run cases • Add new run case • Save run cases to file • Delete run case • Fetch run cases from file • Name current run case • Write forces to file • Execute run case • Initialize variables • Geometry plot • Trefftz Plane plot	• Stability derivatives • Total forces • Body-axis derivatives • Surface forces • Reference quantities • Strip forces • Element forces • Design changes • Strip shear & moment • Options • Hinge moments

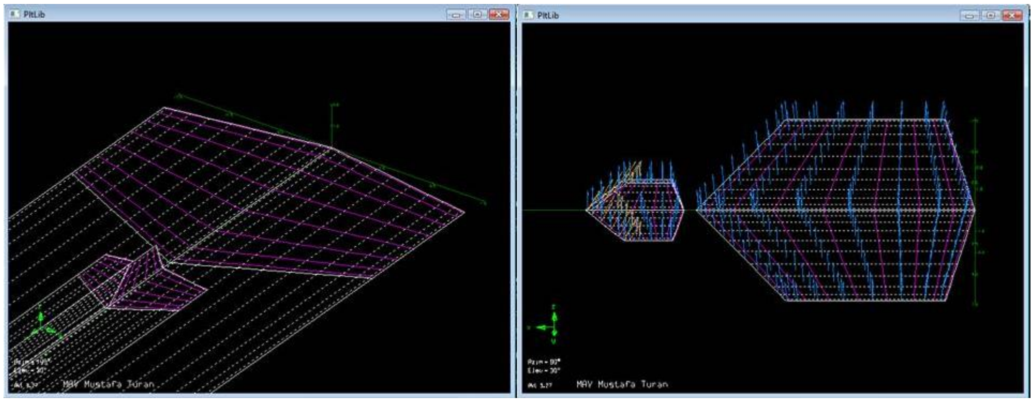
It is a highly complicated process so two batch files that would simplify this process were needed. One of the AVL batch files runs AVL executable, loads xxx.avl, xxx.mass and xxx.run input files and runs the second batch file in order to operate inside the AVL menu. The second batch file opens up the Oper sub-menu, overrides the roll, pitch, yaw rates and flap, aileron, elevator, rudder deflections finally setting the AoA to the desired value. Based on these inputs, the next command in the batch file runs the execution and has the AVL print out the results in the user-defined text file (mtrn_MAV_results.txt in MCDA). Once the results are printed, the AVL filewrapper searches for predefined slots in the results file and fetches the desired data. Results are as follows:

- Run Case Aerodynamic Coefficients: Cl_{tot}, Cd_{tot}, Cd_{ind}, Cl_{ff}, Cd_{ff}, e

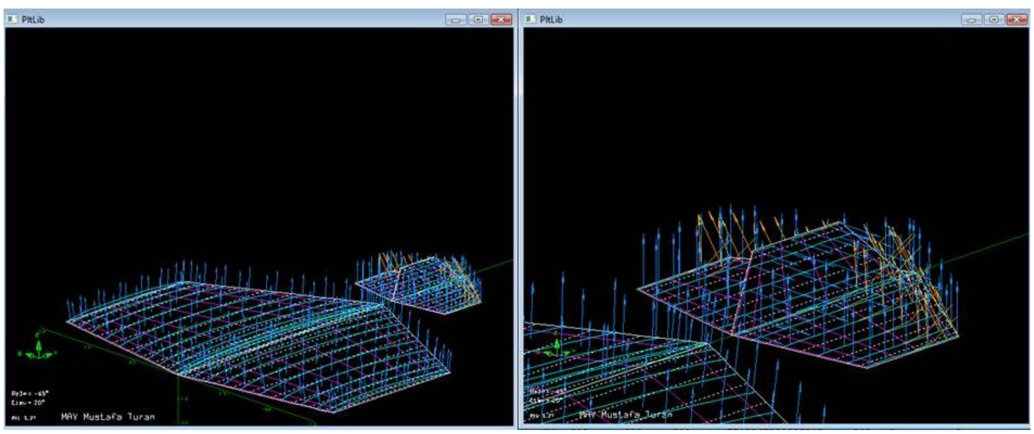
- Control Surface Deflections: Flap, aileron, elevator, and rudder deflections for the desired flight condition
- Stability Axis Derivatives: CL_a, Cy_a, Cl_a, Cm_a, Cn_a, CL_b, Cy_b, Cl_b, Cm_b, and Cn_b are displayed in MCDA but there are other stability derivatives available and filewrapper can be adjusted to the user request. They are CL, Cy, Cl, Cm, and Cn values for the roll rate (p'), pitch rate (q'), yaw rate (r'), flap, aileron, rudder and elevator. Also Trefftz Drag and span efficiency for the flap, aileron, rudder and elevator are displayed and the neutral point is calculated. As seen there are many outputs and a user can tailor the filewrapper to their needs, get data easily and relay it to another component.

After MC runs the AVL filewrapper, all input files must have changed accordingly, so AVL can be run manually with already manipulated input files, i.e. the user does not have to create input files. If the user wants to view the AVL geometry, AVL should be used from the command prompt and the procedures mentioned in the manual should be applied. Figure 3.26 and 3.27 shows some of the functions of the AVL geometry view. These plots were taken after running MC i.e. MC changes the xxx.avl geometry as expected.

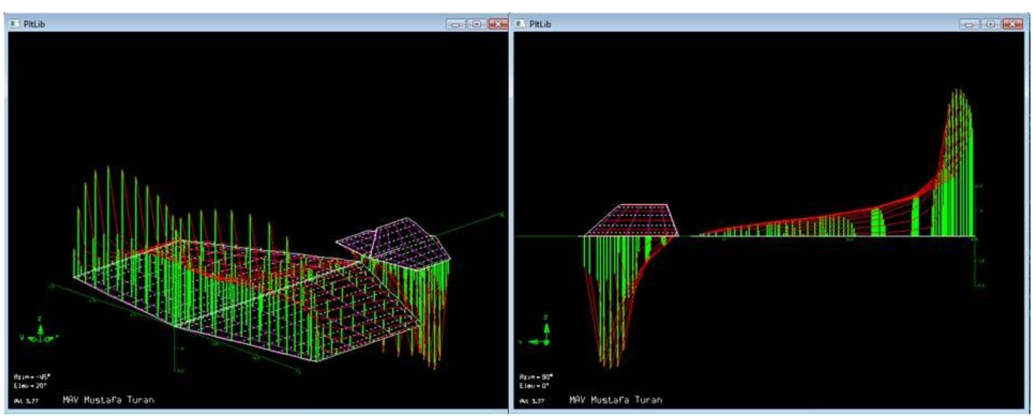
As mentioned in reference [12], difficult-to-model aircraft parameters can be obtained from AVL and Cloud Cap Technologies has used AVL to model almost 20 aircraft and it has in general performed very well. Drela explains the limitations in the avl_doc.tex file which comes with the program [19].



(a) AVL in General

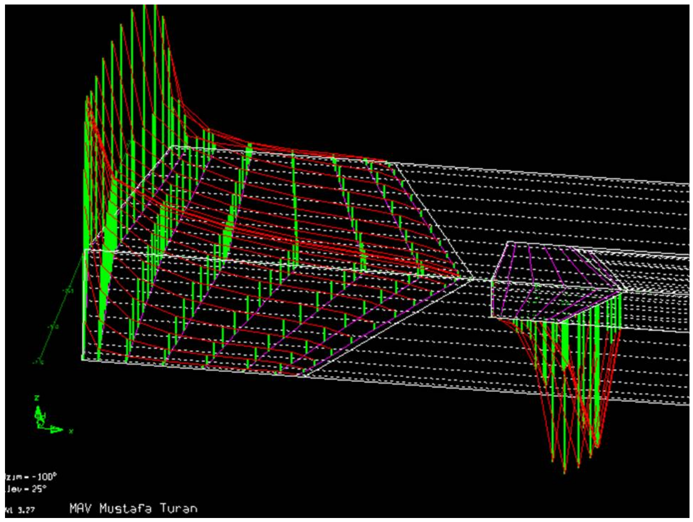


(b) Normal Vectors

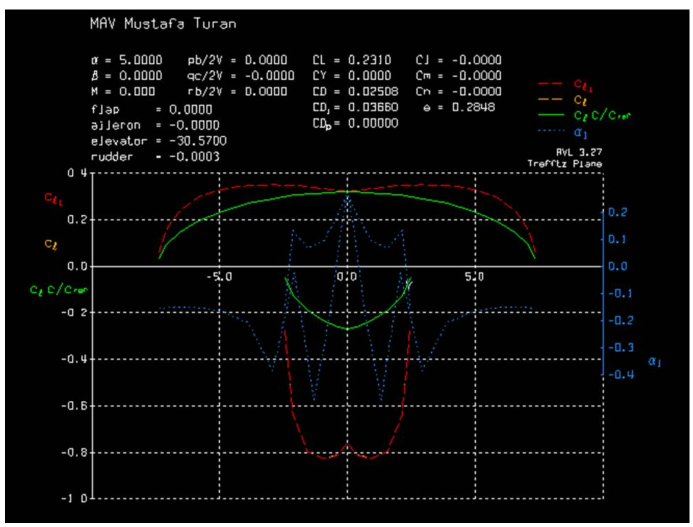


(c) Wing Loading

Figure 3.26   AVL Geometry View Examples 1



(a) Trails and Wing Loading



(b) Trefftz Plane Analysis

Figure 3.27 AVL Geometry View Examples 2

3.3.10 Aircraft Components. MC comes with basic components and shapes for visualization. It has predefined geometric shapes such as block, cone, cylinder, sphere, arrow and some more generic shapes. There are also custom-built component packages such as Aircraft Geometry which is utilized in the MCDA tool. One important thing to mention here before explaining any of the details is that, in general, MC Geometry view is not designed as a computer aided design (CAD) tool. It is intended to give user an intuitive interface to ascertain whether parameters represent the expected values, either input or output. Aircraft Geometry component in MC was built in that sense as well. It has nose section, mid-section, aft-body components as the fuselage of the aircraft, wing or multisection-wing components as the main wing, horizontal wing, vertical wing or canard types. They are represented in a primitive way in the geometry view as seen in Figure 3.28. For example, when b and c_{root} is increased, immediately after that the changes can be observed, but if the wing profile is changed, it can't be observed in the geometry view. The same type of restrictions apply to the fuselage. Only elliptical or circular types of body shape can be entered. Therefore, wing volume and wetted surface areas cannot be calculated properly without having the complete geometry. These type of limitations were also reviewed by Dittmar [16] and he wrote several MATLAB codes to allow the calculation of wing volume and surface areas and created super-elliptical fuselage shapes. He also evaluated the General Geometry Generator (GGG) version 2.0 for potential use and inclusion into MC.

Even with all those restrictions, Aircraft Geometry Component package is very useful for a generic aircraft in order to manipulate the respected data easily and visualize the work. Now properties that can be input in Aircraft Geometry components will be presented.

- **Nose:** Dive angle, geometry (in axis system), number of cross sections (when changed, no effects were observed by the author), length, radius-1 and radius-2, shoulder and tip angle.

- **Mid-Section:** Dive angle, geometry (in axis system), length, number of cross sections, radius-1 and 2 for defining the first cross section, radius-3 and 4 for defining the last cross section (either ellipse or circle).

- **Aft-Body:** Geometry, angle-1 and 2, length , number of cross sections, radius-1 and 2 for defining the first cross section, radius-3 and 4 for defining the last cross section (either ellipse or circle).

- **Wing:** AR, $(t/c)_{tip}$, $(t/c)_{root}$, S_{ref}, dihedral angle, number of cross sections, c_{root}, c_{tip}, b, Λ, λ, twist angle, type (4-wing, 5-htail, 6-vtail, 7-canard)

- **Multi-Wing Section:** $(t/c)_{tip}$, c_{tip} , number of sections (operational with some restrictions), b, Λ, twist angle. These values are input as array of numbers up to 5 different cross sections.

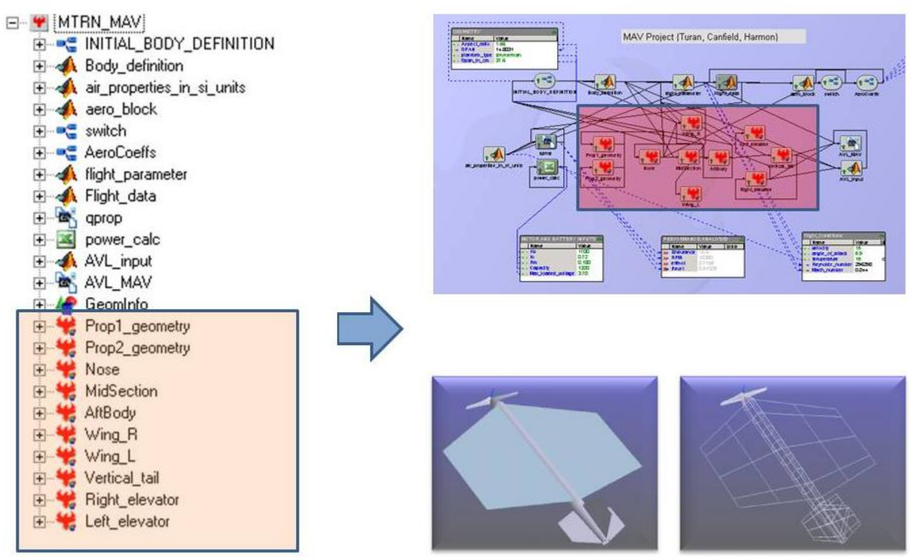


Figure 3.28 Aircraft Geomerty Component Overview

As seen from the component properties, a generic aircraft can be created in MC with some limitations. It is possible to enter each value by hand or by linking with Link Editor (one of the most powerful features of MC). Every aircraft component in the MCDA tool is linked together and once the MCDA user defines the initial

parameters via the Geometric Properties Calculator, it will automatically calculate linked components and will display in geometry view.

3.3.11 Propeller Geometry. The propeller geometry was created with the Multi-Wing Section component by the author, since there is no specific component for propulsion. It is less detailed than QPROP, which has highly sophisticated propeller geometry. Therefore, the propeller geometry component is not capable enough to cooperate efficiently with QPROP with only five cross sections embedded in the multi-wing section component. As mentioned before, the MC geometry view was not designed as a CAD tool; some script and MATLAB plug-ins can alleviate this problem. As a second option, manually changed input files for QPROP can be used before running the MCDA. The second method was used in the MCDA tool (due to time restriction), although it is controversial to the philosophy behind using MC. A user should be able to change all the parameters within the MC environment. Once the database input files in QPROP are standardized, the data will be easily manipulated in the MC environment in future research. As an example, propeller geometries in QPROP database propeller files have different numbers of cross sections. When a filewrapper is generated, the user is limited to the number of inputs stated in filewrapper and if another propeller input file has different number of parameters, then it will not be possible to use the already created filewrapper in MC, because it will not be able to match the pre-defined and actual input file variables.

4. Results and Analysis

This chapter includes the evaluation of experimental data interpolation performance of the MCDA tool, validation of the MCDA tool using BumbleBee MAV, a coaxial prototype analysis with related changes to the original tool, and the QPROP tool performance analysis (without validation). Some of the results are presented in the related Appendices.

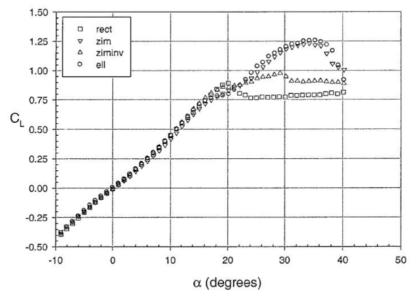
4.1 Experimental Data Interpolation/Evaluation

This section includes the evaluation of the experimental data created by the Aero-Block (experimental interpolation) component in the MCDA tool. Initially, it is shown that interpolation of Torres and Mueller's [47] experimental data was accomplished with a properly created data set. In the second part, Marek's experimental data was compared with the outcome of the Aero-Block and DATCOM component.

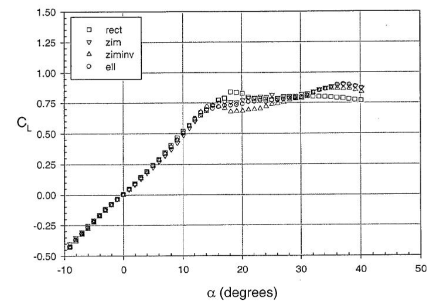
4.1.1 Aero-Block Performance Evaluation. Figure 4.1 has the comparison of the actual experimental data created by Torres and Mueller and aero-block generated data. Aero-block generated data should match the experimental data and moreover, it should correctly interpolate the data.

Some parametric trade studies were run in MC for each planform separately. As previously mentioned, in order to get results from the aero-block, user has to define AR, Re and AoA. The parametric study tool has a plotting feature, but to show all planforms in a single plot, the results were exported into the MC xxx.cvs file and then plotted using MATLAB.

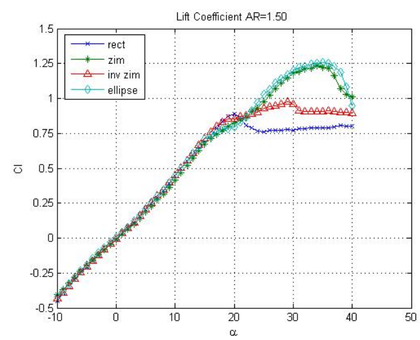
Figures 4.1(a) and 4.1(b) are the actual experimental C_L data of all planforms with AR=1.5 and AR=1.75 respectively at Re=100K which were picked from the final report of Torres and Mueller [47] for comparison. Figures 4.1(c) and 4.1(d)



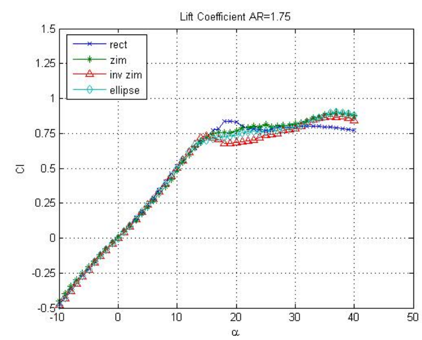
(a) Lift Coefficient AR=1.50 [47]



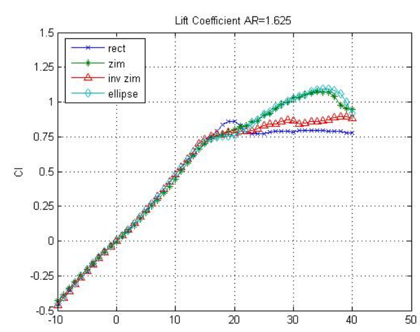
(b) Lift Coefficient AR=1.75 [47]



(c) Lift Coefficient AR=1.50



(d) Lift Coefficient AR=1.75



(e) Lift Coefficient AR=1.625

Figure 4.1 Experimental Data [47] and MCDA Aero-Block Generated Experimental Data

are the aero-block generated experimental data plotted in MATLAB. As seen in related figures, they match the experimental data. Finally, Figure 4.1(e) shows the interpolation of the C_L data of inverse Zimmerman planform with AR=1.675 at Re=100K and interpolation occurred between AR=1.5 and AR=1.75. Consequently, it produced the results as expected. See Appendix B for 3D MCDA interpolation results for the inverse Zimmerman planform.

4.1.2 Experimental Data Cross-Validation. Marek [38] conducted some wind tunnel experiments similar to Torres [47] and created a database for his design and optimization instead of using VLM or some sophisticated CFD codes. Therefore, he used a method based directly on data from wind tunnel experiments. In his wind tunnel experiments, the method was validated for higher Reynolds numbers than described in Torres [47]. The present research focuses on the topic using same type of approach but with a multidisciplinary tool. Marek's results are compared to MCDA and DATCOM results in Figure 4.2. Experimental setup included inverse Zimmerman planform, 30 cm b, 3.1% t/c, AR=1.66 and Re=140K.

The b, AR and type were entered in the MCDA tool and parametric study was run to find the velocity that would give the Re=140K at $T_{celc} = 15$. After finding u= 8.886 m/s and M=0.026, two separate parametric trade studies were run for Aero-Block and DATCOM to compare the aerodynamic coefficients. Figure 2(a) has Marek's $C_L,\ C_D\ and\ L/D$ results for the given configuration. Figure 2(b) has $C_L,\ C_D\ vs\ \alpha$ of the MCDA and DATCOM. Figure 2(c) has $L/D\ vs\ \alpha$ of the MCDA and DATCOM. For the current case, MCDA interpolated its own database based on Torres' work and DATCOM was run by MCDA via assumptions, whereas Marek's results are directly from wind tunnel tests. The findings are:

- MCDA $C_{L_{max}}$ was 18 % higher in magnitude than Marek's and was not within the error bars. However, stall AoAs almost matched. DATCOM $C_{L_{max}}$ was

6% higher in magnitude than Marek's being within the error bars. Stall AoAs had the same tendency.

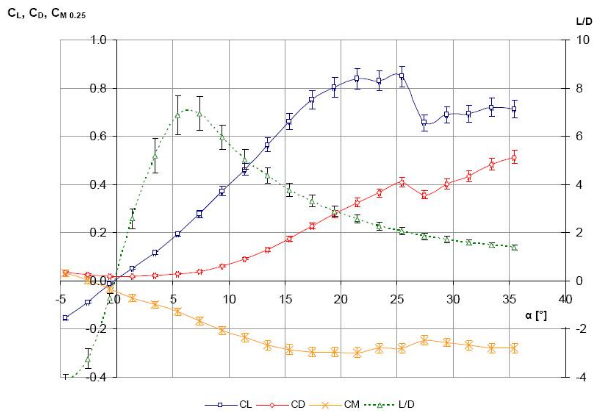
- MCDA C_D was slightly higher after 12° of AoA but DATCOM C_D was within the error bars up to stall AoA.
- L/D_{max}=7.2 occurred at about 6° AoA in Marek's results. MCDA results were uneven due to interpolation and MCDA L/D_{max}=7.4 at 4° AoA. After polynomial curve fitting, MCDA L/D_{max} was about 7 at 5° AoA (within 3 % of Marek's results). DATCOM L/D_{max}, unexpectedly very low, was about 5.2 at 6° AoA (within 28 % of Marek's results).

Overall, MCDA results are believed to be in good approximation with all the assumption taken. DATCOM produced reasonably good results for a 30 cm-wing. The MCDA L/D ratios are very sensitive at lower AoAs so the user has to be cautious about the jumps in L/D and take appropriate action by deleting invalid run parameters.

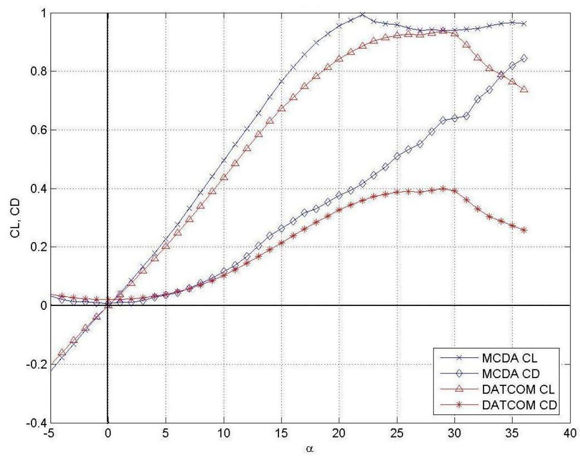
4.2 Validation of The MCDA Tool Using BumbleBee MAV Prototype

Marek also designed a code to optimize the wing geometry of the MAV and his optimization code was expected to find the best wing geometry within the constraints set by the user. He created a prototype called "BumbleBee" with his predefined geometry parameters. BumbleBee had L/D_{max} of 5.66, b of 376 mm, Hacker $A10-9L$ electric motor, 1320 mAh Lipo battery and $HackerX7$ $10A$ speed controller with generic 6x4 propeller. Its endurance was calculated by MotoCalc as 23 minutes at 85 % throttle settings. The total weight of the BumbleBee was 305 g and it had an inverse Zimmerman wing planform. Some flight tests were carried out and for an approximate cruise velocity of 15.4 m/s, maximum flight time logged was 19 minutes. The cruise speed and endurance were close to his initial design parameters.

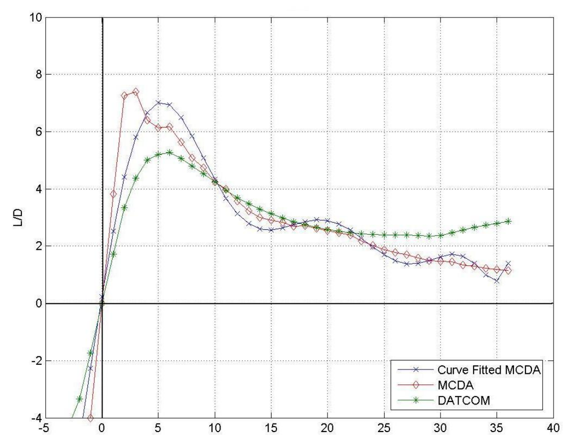
The BumbleBee geometry and the other parameters were matched and run in the MCDA. Motor properties for the same motor [3] and the same battery parameters



(a) Marek's Test Result [38]



(b) $C_L,\ C_D\ vs\ \alpha$



(c) L/D vs α

Figure 4.2 Comparison of Test Results [38] to Aerodynamic Coefficients Generated by MCDA

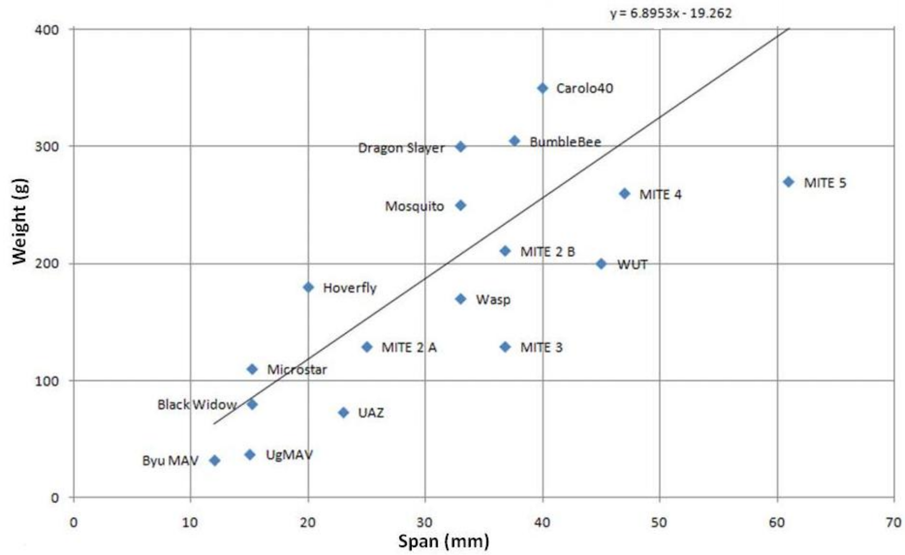


Figure 4.3 MAV Weights vs Span derived from [51, 38, 46]

were applied. The actual weight of the vehicle was higher than the estimated weight according to Figure 4.3, so it was set to the actual weight of BumbleBee in the MCDA tool.

The propeller geometry chosen from the QPROP database, cam6x4 (Graupner CAM 6x4), was selected as the closest geometry in size. A factor of 1.2 is used to account for the additional power loads (e.g. avionics, servos, transmitter, wire resistance, etc.). Figure 4.4 shows the steps followed in the evaluation of the BumbleBee MAV. The flight conditions, geometry and propulsion data were set based on the parameters mentioned above and initial geometry was confirmed by looking at the Geometry View of MC. The L/D vs AoA case was run by the parametric study tool in MC and L/D_{max} value was 4.5 at 9° AoA and 12 m/s velocity. The geometry used by DATCOM was also checked for discrepancies with MC. The matching planforms are shown at the right lower corner of Figure 4.4. Since Reynolds number was higher than the database limits, extrapolation was used within the code. The $C_L(C_D)$ curves are compared in Figure 4.5 (Marek's result on the left and MCDA result on the right) and results are a little different than each other which is expected

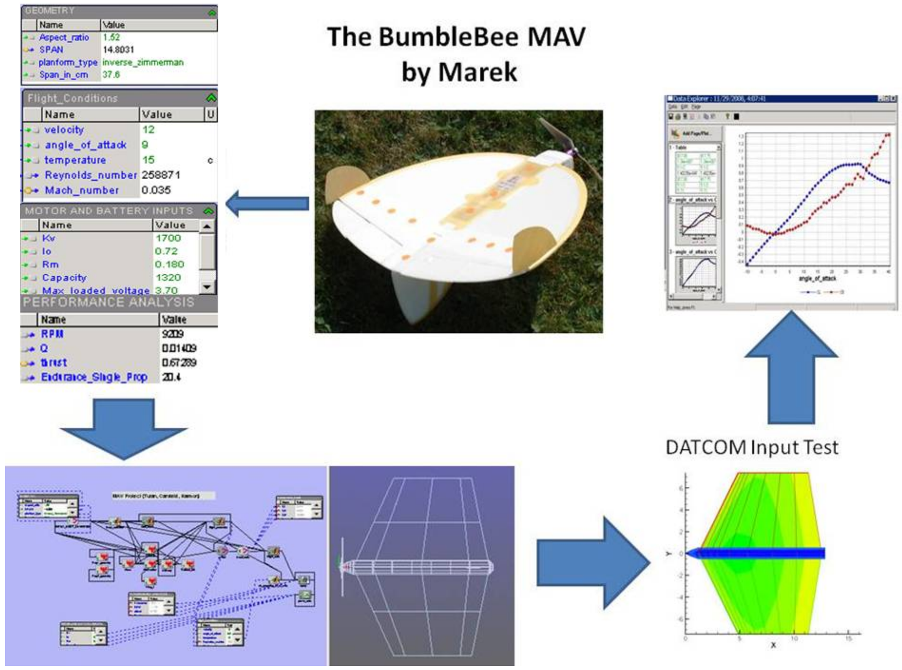


Figure 4.4 BumbleBee MAV Evaluation in the MCDA Tool

since Reynolds number was higher then the MC database limits and extrapolation was used within the code. Finally endurance of 20.4 minutes (within 7.4 % of the maximum logged flight time of the BumbleBee MAV) at 9210 RPM with a thrust of 0.672 N was found. Consequently, endurance result closely matched the Bumblebee result.

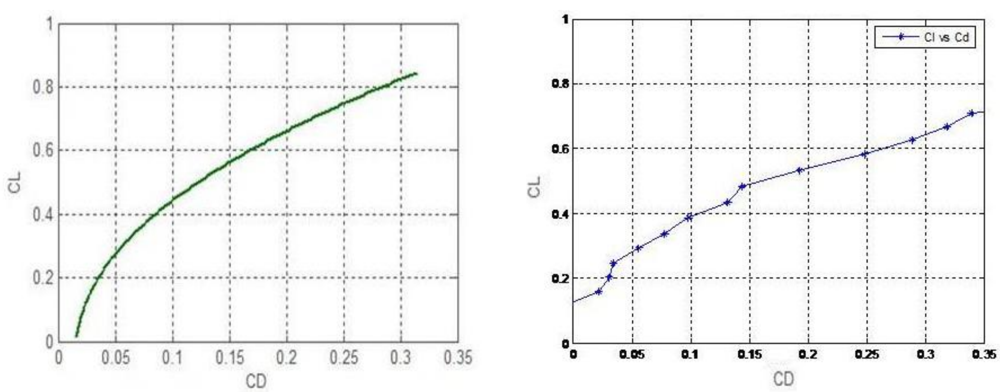


Figure 4.5 BumbleBee and MCDA C_L vs C_D

4.3 Coaxial MAV Prototype

In this section, a coaxial MAV prototype in the MCDA tool is evaluated after some modifications to the single-propeller version.

4.3.1 Coaxial MCDA Tool. The proposed mission type had two types of mission profiles: forward flight and hovering flight. The MCDA tool was created for single-propeller MAVs. It is modified to evaluate the hovering flight phase. Based on the literature review, counter-rotating propellers would provide torque control on the vehicle while hovering. There are some state-of-art R/C vehicles with additional controls. As an example, Blade mCX is one of the off-the-shelf micro helicopters with coaxial counter-rotating blades. It has a total weight of 28 g (1.0 oz). It delivers flight times of 6 to 8 minutes, while full, 4-channel control provides the precision needed for flying in tight indoor spaces. It has a unique 5-in-1 control unit with combination of main motor electronic speed controls, mixer, gyro, servos and receiver [1]. It also has two additional motors to move the swash plate up/down and left/right which provides exceptional maneuverability. A similar approach was imagined for the proposed mission profile in the hovering phase.

The MCDA tool was modified (Figure 4.6) to have two counter-rotating propellers. As a starting point, the same type of propeller and motor as in Bumblebee was used in the evaluation. The modifications are:

- Another propeller and a shaft were added in the geometry section for visual validation.
- Two additional components were created under the name of Forward and Hover Coaxial Propulsion. Each propeller has its own QPROP filewrapper and they feed the Power Calculation spreadsheet together for the endurance.
- Modified "Flight Data" component provides thrust (T) and velocity (u) for each of the propellers i.e. separate QPROP components. Two subcomponents

were added to "Flight Data" component: "hovering data" and "forward flight data". For the hovering phase of flight, $T_{req_{total}}=W_{appx}$ and u=0. For the forward flight phase $T_{req}=\frac{W_{appx}}{(L/D)_{max}} * \frac{g}{1000}$ for a given u.

QPROP, Flight Data, Power Performance Calculator and Propeller Geometry components have exactly the same variables and are explained in Chapter 3.3 in detail .

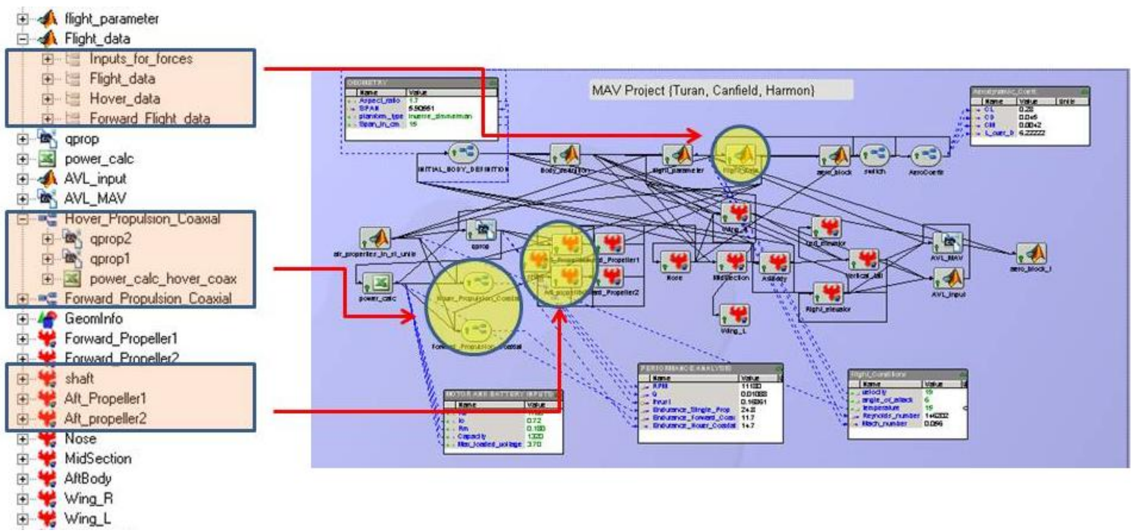
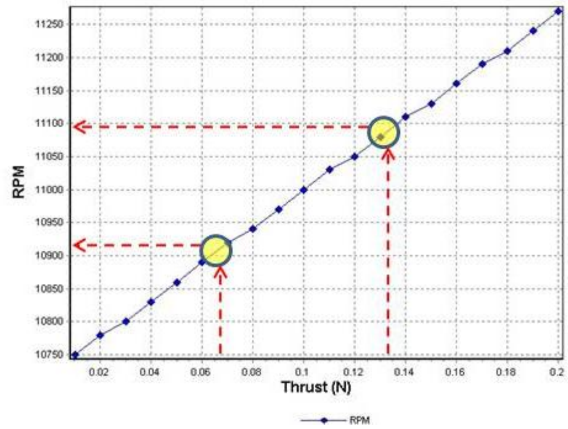


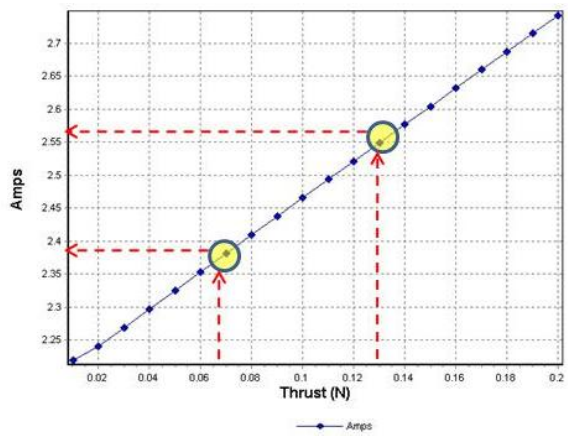
Figure 4.6 Modifications to MCDA for Coaxial Configuration

4.3.2 Analysis of the Coaxial Prototype. After modifications were applied, a MAV with inverse Zimmerman wing, 15 cm b, AR=1.7 and W_{appx}=84 g (based on the Figure 4.3) was evaluated in the coaxial MCDA tool. It was found that it should fly at 19 m/s in forward flight with L/D=6.22 and 6° AoA. Single-propeller forward flight, coaxial-propeller forward flight and coaxial-propeller hovering flight results are presented in Figure 4.7. Coaxial results represent the single motor parameters.

For coaxial hovering and forward flight, it is assumed that thrust is shared by two propellers equally. In the trade study, coaxial configurations were supposed to have less endurance due to the extra load by the second motor for the propulsion, and additional motors and servos for maneuverability. The coaxial MCDA tool estimated

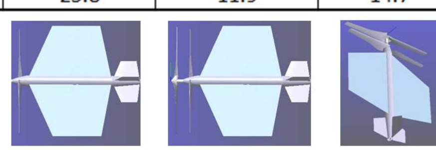


(a) RPM vs Thrust at u=19 m/s



(b) Amperes vs Thrust at u=19 m/s

	Single Forward	Coaxial Forward	Coaxial Hover
RPM	11090	10910	4302
Q	0.01032	0.00928	0.00542
Pshaft	11.98	10.59	2.44
Volts	6.982	6.842	2.834
Amps	2.5566	2.3713	1.6841
Effmot	0.671	0.6529	0.5112
Thrust	0.13266	0.06633	0.41272
Velocity	19	19	0
Endurance	25.8	11.9	14.7



(c) MCDA Results

Figure 4.7 Different Flight Mode Results

that the coaxial forward flight had less endurance than the coaxial hovering for the current configuration.

4.4 QPROP Performance Analysis

In the MCDA tool, the QPROP component was restricted to T_{req} and u. In order to evaluate the QPROP performance, a separate component was created under "Evaluation/Propulsion" with complete capabilities of the actual single-point-run QPROP, instead of multi-point-run because MC can run parametric studies itself. Three different parametric studies were conducted in the MCDA tool for evaluation of the QPROP program.

During the run cases, the fluid constant and the motor input files were kept the same. QPROP requires a detailed description of the propeller geometry and blade airfoil characteristics but a user can create his own sophisticated propeller properties as mentioned in Reference [17]. Therefore, to keep the analysis simple, the propeller file (geometry and number of blades) was manipulated in parametric studies. The first goal was to enlarge the propeller geometry by increasing the radius (r), therefore chord lengths (c) increase accordingly, at each station but blade angles (β) remain the same. The second goal was to change the number of blades during the parametric studies. The Graupner CAM 6X3 folder propeller properties [17] were changed with the following assumptions.

In the propeller file, $C_{L_0},\ C_{L_a},\ C_{L_{min}}\ and\ C_{L_{max}}$ parameters stay the same when the propeller is enlarged with only radius and chord (blade angle at each cross section remain the same) via the following equations:

$$r_{ratio} = \frac{r_{user}}{r_0} \tag{4.1}$$

$$r[7] = r_{ratio} * [r1,\ r2,\ r3,\ r4,\ r5,\ r6,\ r7] \tag{4.2}$$

$$c[7] = r_{ratio} * [c1,\ c2,\ c3,\ c4,\ c5,\ c6,\ c7] \tag{4.3}$$

$$\beta[7] = [b1,\ b2,\ b3,\ b4,\ b5,\ b6,\ b7] \tag{4.4}$$

After applying these assumptions, three different parametric studies were conducted by changing the radius and number of blades of the propeller. Units were kept the same as in the QRPROP input files.

4.4.1 T and RPM Relation While Changing Number of Blades. In this setup, u was set to zero (i.e. static platform measurement). T was increased from 0.2 N to 2.0 N with a 0.1 N increment and the number of blades was increased from 2 to 5 with 1 increment in the parametric study tool. A total of 19 runs were conducted in 2 minutes. Finally as seen in Figure 4.8, it is found that as T_{req} increases, RPM increases. As the number of blades increase, RPM decreases for the same T_{req} case.

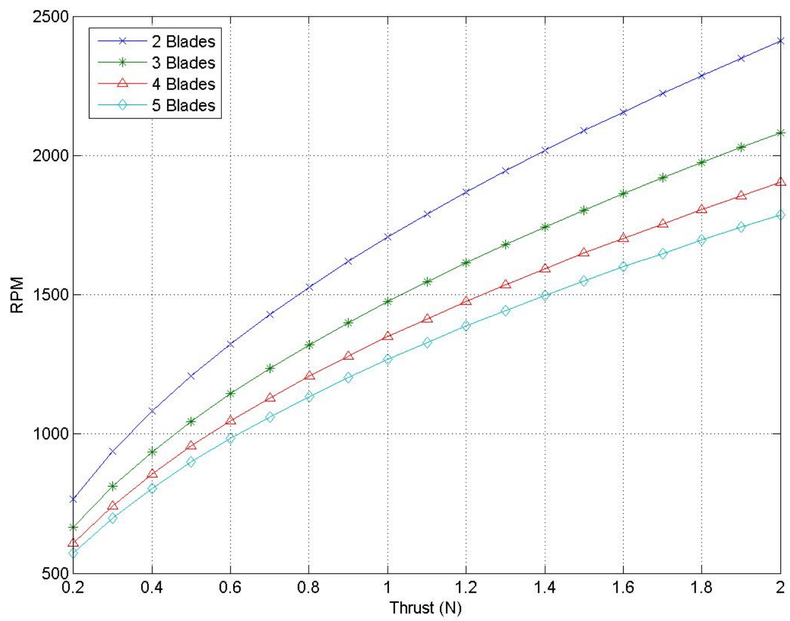
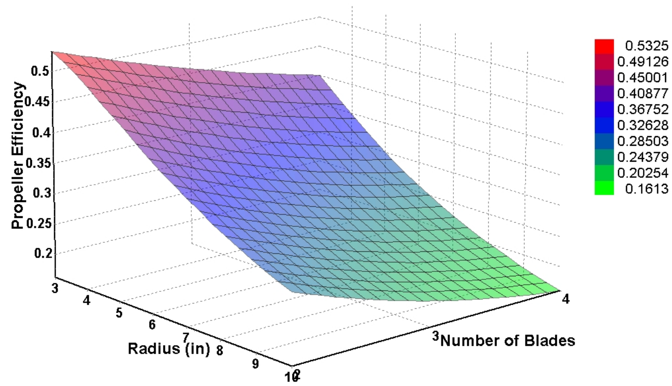


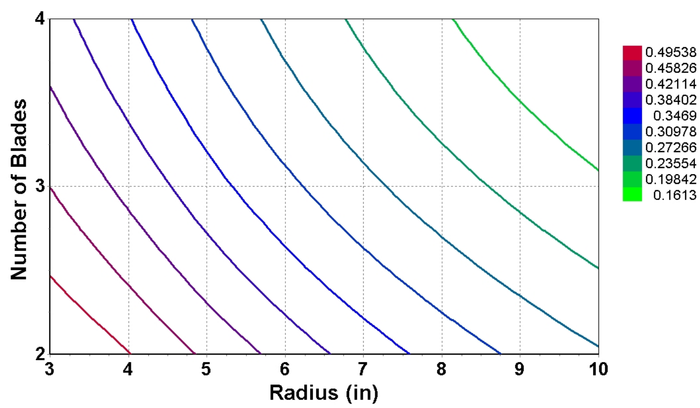
Figure 4.8 RPM vs T

4.4.2 Propeller Parameters While Changing Number of Blades and r. In this setup, u was set to 10 m/s and T_{req}=0.4 N. Based on the assumptions, the radius of the propeller was increased from 3 in to 10 in with an increment of 1 in

and the number of blades was increased from 2 to 4 with an increment of one. A total of 24 runs were conducted in 4 minutes. The C_T, η_{prop}, I, V, RPM parameters were compared in the parametric study tool. Results for the η_{prop} are presented as an example in Figure 4.9. The C_T, I, V, RPM results are presented in the same fashion in Appendix C.1.



(a) 3D η_{prop} vs r and Number of Blades

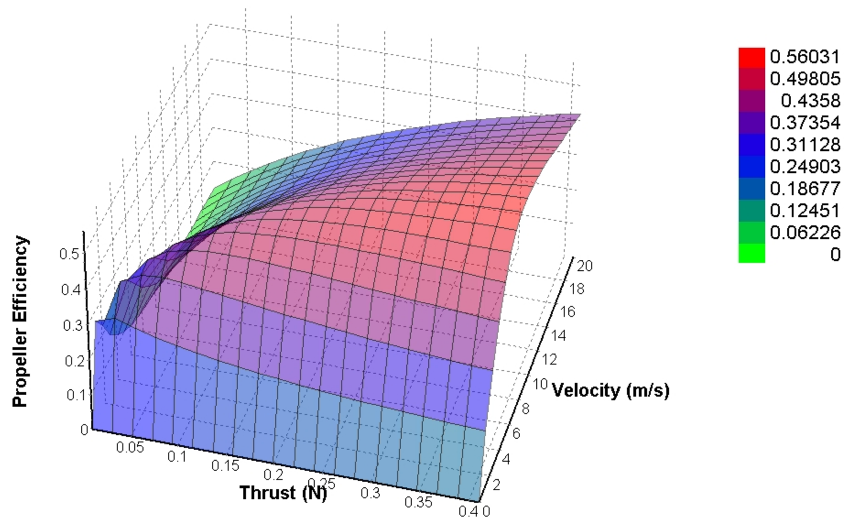


(b) 2D η_{prop} vs r and Number of Blades

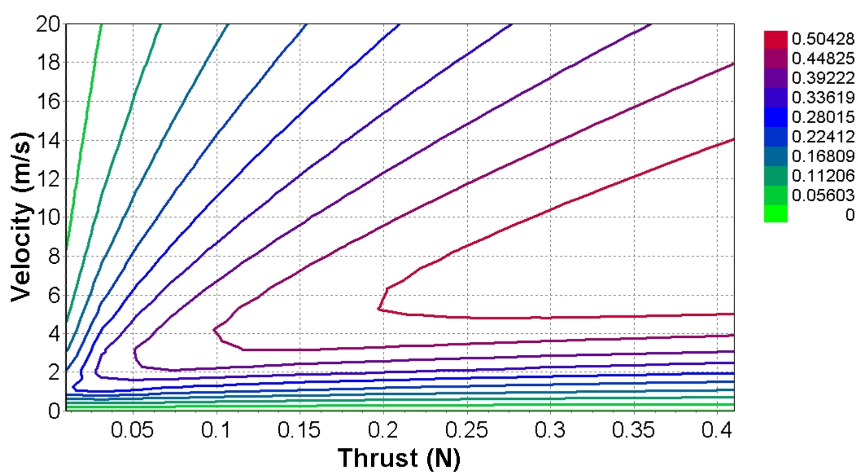
Figure 4.9 Change in η_{prop} with r and Number of Blades

4.4.3 Propeller Parameters While Changing T and u. In this setup, the T_{req} was increased from 0.01 N to 0.4 N with an increment of 0.02 and the u was increased from 0 m/s to 20 m/s with an increment of 1 m/s. A total of 441 runs were conducted in 45 minutes. The C_T, Q, η_{prop}, I, V and RPM parameters were compared

in the parametric study tool. Results for the η_{prop} are presented in Figure 4.10. The C_T, Q, I, V and RPM results are presented in the same fashion in Appendix C.2.



(a) 3D η_{prop} vs T and u



(b) 2D η_{prop} vs T and u

Figure 4.10 Change in η_{prop} with T and u

5. *Conclusions and Recommendations*

This chapter includes conclusions, limitations of the research and recommendations.

5.1 Conclusions

In recent years, research in MAV is attracting people from all around the world. Several universities have been involved in MAV research and people are still researching the physics behind different kinds of MAVs. It requires different disciplines to design a MAV, like any other vehicle, but the number of constraints are increasing due to the small scale of MAVs. Battery or any source of energy, propulsion, component weights, MAV building materials, low Re number effects on aerodynamics, unpredictable propeller-induced flow effects over the body and surfaces, gust effects etc. are restricting the MAV capabilities, therefore the MAV missions.

The current research focused on a multidisciplinary approach to fixed-wing MAVs in a very suitable integration environment, called ModelCenter (MC). A tool, fixed-wing MAV Conceptual Design and Analysis (MCDA), was created within the MC environment. A single-propeller and a coaxial MAV were evaluated with the MCDA tool. Due to the unique characteristics of the LAR wings at low Re numbers, experimental data, although it's limited, was integrated into the model to supplement DATCOM. Some of the R/C community approaches such as QPROP were applied in the propulsion part and evaluated in Chapter 4.4 and Appendix C. For stability and control, AVL was integrated into the model but not with its entire capabilities. Working on each of these areas individually takes too much time but when the related software programs were configured properly to work within the MC, work load and data processing time decreased and quick data evaluations could be made easily. Data validation of the tool was made by comparing similar research on a fixed-wing MAV with the MCDA tool outputs. However, the tool itself is not a generic tool to

evaluate all types of fixed-wing MAVs and has some limitations due to the integrated component limitations, limited experimental data, limited database, etc. Not all of the variables for each of the software programs were used in the current research and to have a better analysis, it is critically important that a through understanding of all the variables associated with each of the major software programs is needed. Also, the tool itself is not at the component level, though it can be done easily with the guidance of the current research. It was successfully shown that some sophisticated software programs, that will help analyze the conceptual design, can be integrated and evaluated in the MC environment. Also, some of tools that were revealed in Appendix A can be explored for extending the MCDA capabilities further. ModelCenter has proven itself to be a very good environment for conceptual design, although Aircraft Geometry component was limited to certain shapes. It is believed that optimization capabilities and various plug-in components will add a substantial power to the MCDA.

5.2 Limitations of Research

The MCDA is limited to fixed-wing MAVs. Due to lack of aerodynamic data on the LAR wings at low Re numbers, experimental data supplemented DATCOM was used in aerodynamic analysis. Since the area of interest was the low Re numbers, we were restricted to the experimental data of Torres'. The experiment was conducted with four different flat-plate wing planforms, seven different AR and two different Re numbers. Besides being limited to experimental data, DATCOM also limits the user for certain geometric shapes and some assumptions had to be made to correlate with the experimental data. For QPROP, extensive research is needed for the propeller and motor database. Although it has limited motor types, any motor model can be coded in SUBROUTINE MOTORQ (in motor.f). Moreover for non-electric motors, the voltage (V), passed to MOTORQ, can represent any suitable power-control variable, e.g. throttle setting, fuel flow rate, etc. QPROP

has two propeller file formats: simple and advanced propeller input files. A user can create his data by the guidance shown in the manual which will boost the QPROP capabilities when combined with the MC parametric study tool. AVL is a very good software in terms of being able to create unconventionally-shaped vehicles, unlike DATCOM. It can also calculate aerodynamic coefficients as well as stability and control derivatives. There are some applications of AVL in the simulation world but it has some limitations as mentioned in its manual. The AVL geometry file was integrated successfully and it represents the same geometry in MCDA, although it is possible to configure many different shapes. Mass and run files are needed to be explored in detail, and it requires a considerable amount of time and expertise on AVL.

5.3 Recommendations

The current research has revealed some software programs and integration procedures for them into MC for the proposed effort. It is important to utilize all capabilities of each of the tools that were integrated. Therefore, possible areas of improvement are:

- **DATCOM:** In the MCDA tool, the shape of the body segment is restricted to cylindrical shape due to predefined MC Aircraft components. However in DATCOM, besides cylindrical shape, a user can input cambered bodies of arbitrary cross section by specifying the BODY namelist optional inputs. DATCOM can compute static longitudinal and lateral stability. It can also compute dynamic derivatives but the solutions are provided for basic geometry only and not all of the dynamic derivatives are calculated for each combination of vehicle configuration and speed regime because of DATCOM limitations. For the dynamic stability, the effects of high-lift and control devices are not recognized either.

- **QPROP:** This software is fully functional in the MCDA tool if the input files are present. Electrical or Motocalc database have variety of motors, batteries, ESCs, etc. A similar MATLAB or Excel database could be created for micro-motors, batteries and propellers that would cooperate with QPROP.

  QPROP is well-documented and a user can create his own propeller input file by the guidance of the user manual which may require integration of Drela's Xfoil or a similar software into the MCDA. It also possible to change the subroutine of QPROP for different type of motors.

- **AVL:** The AVL geometry file was integrated into the MCDA tool but in order to evaluate and utilize the tool, mass and run files need to be constructed properly. After this step, AVL filewrapper has to be edited so that it will manipulate all of these three files (geometry, mass, and run) for MC operations. AVL and other VLM software programs have to be analyzed for LAR wings at low Re Numbers.

The MC environment is a very flexible environment. There can be some add-on's to the current MCDA tool. Appendix A may be a starting point for that purpose. In any case, a stable tool, either a software or experimental research that would cover the entire flight regime of the MAV, is needed for aerodynamics. The induced flow (prop wash) effect on the body and other surfaces has to be taken into account as well. Momentum theory can be applied to figure out the propeller flow field. Motor on/off experiments can be conducted to see the effects of the induced flow over the body and surfaces but will require great amount of time and effort. In addition to those, there is another important issue that the speed of the wind gust may be on the same order of magnitude as the overall flight speed of a MAV and maintaining smooth flight can be a challenge for either a R/C pilot or an AFCS. Therefore, gust tolerance modeling tool is a must in the proposed effort. A MATLAB m-file specifically created for this purpose can easily be integrated into MC. It is recommended to have people work on various disciplines separately and have them

combine their work at regular intervals, to be compatible, in an environment such as ModelCenter.

Appendix A. Survey of Tools

This Appendix lists different tools (Airfoil, Propeller and CFD/Aerodynamics Analysis) that could be used for MAV conceptual designs.

1. D-calc, Christian Persson/Helmut Schenk (English & German)

   www.yahoogroups.com/group/D-calc

2. MM_calc, English, Louis Fourdan (freeware)

   electrofly.free.fr

   -> tlchargements (link at top of page)

   -> moteurs

   -> MM_calc

   -> English or French version

   Help/discussion/announcements/bug-reports/pats-on-shoulder:

   http://www.rcgroups.com/forums/showthread.php?t=583327

   Links to MM_Calc derivatives:

   Scorpion_Calc, Motrolfly_Calc, Dualsky_Calc, Aero-nuts_Calc, Himax_Calc

   http://www.rcgroups.com/forums/show...714#post8760714

3. Motocalc ($)

   www.motocalc.com

4. Elektro-Antrieb (German only, ($)

   www.geck-elektroantrieb.de

5. P-calc (freeware)

   brantuas.com/ezcalc/dma1.asp

6. E-Calc

   www.slkelectronics.com/ecalc/index.htm

7. Mumtats (freeware), RCGroups user 'vintage1'

   http://www.rcgroups.com/forums/showthread.php?t=233250

8. Rod Badcock's thrust-, prop- and motor-calculators (freeware)

   www.badcock.net

9. Peak efficiency (freeware)

   www.peakeff.com

10. WebOcalc and PowerCalc (Free Open Source Software)

    flbeagle.rchomepage.com

    -> software

    Help/discussion/announcements/bug-reports/pats-on-shoulder:

    http://www.rcgroups.com/forums/showthread.php?t=930018

11. Adam One Motor/Prop calculator (freeware)

    www.adamone.rchomepage.com/calc_motor.htm

12. Thrust calculator (freeware)

    www.lcrcc.net/thrust_calc.htm

13. Thrust calculator (freeware)

    www.gobrushless.com/testing/thrust_calculator.php

14. 'Propellor Calculator' by Helmut Schenk (freeware, English & German)

    www.drivecalc.de

    -> propellor calculator (bottom of page)

15. Prof. Mark Drela's prop calculator

    web.mit.edu/drela/Public/web/qprop

16. Jim Banner's (user 'jrb') calculator

    http://www.rcgroups.com/forums/show...hmentid=1621267

17. The math behind calculators and motors:

    http://www.rcgroups.com/forums/showthread.php?t=185271

18. FanCalc (freeware)

    http://www.s4a.ch/eflight/fancalc_e.htm (English & German)

19. MotorCalc (freeware)

    http://www.s4a.ch/eflight/motorcalc.htm (German)

20. CompuFoil3D

    http://www.compufoil.com/index.shtml

21. Profili

    http://www.profili2.com/eng/default.htm

22. XFOIL

    http://web.mit.edu/drela/Public/web/xfoil/

23. Athena Vortex Lattice

    http://web.mit.edu/drela/Public/web/avl/

24. Eppler

    http://www.pdas.com/eppler.htm

25. MeshPilot:A 2D airfoil mesh CFD analysis tool.

    http://www.shore-cfd.com/html/shore_cfd_-_meshpilot.shtml

26. CRCCsim: A Model-Airplane Flight Simulation Program

    http://crrcsim.sourceforge.net/

27. Adams/Aircraft

28. Java Foil

    http://www.mh-aerotools.de/airfoils/

29. Java Prop

    http://www.mh-aerotools.de/airfoils/

30. Java Pipe

    http://www.mh-aerotools.de/airfoils/

31. Processing of Propeller Geometry

    http://www.mh-aerotools.de/airfoils/

32. Determination of the Aerodynamic Center and
    the Center of Gravity of Planforms

    http://www.mh-aerotools.de/airfoils/

33. PlaneView

    http://www.mh-aerotools.de/airfoils/

34. Cloudcaptech

    http://www.cloudcaptech.com/download/Piccolo/Aircraft%20Modeling%20Tools/

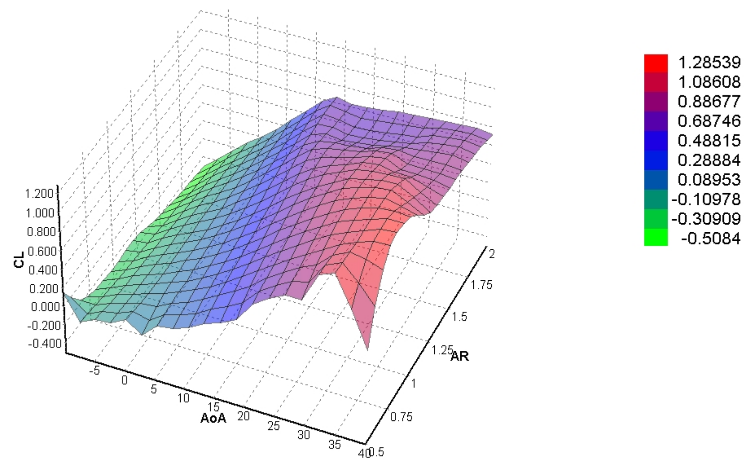
35. Advanced Aircraft Analysis (AAA)

    http://www.darcorp.com/Software/AAA/

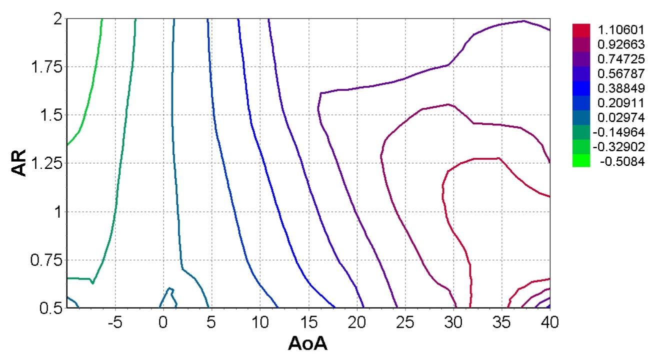
Appendix B. Experimental Data Interpolation in the MCDA Tool

This Appendix has experimental [47] and interpolated aerodynamic coefficients of the inverse Zimmerman planform.

B.1 Inverse Zimmerman at Re=70K [47]

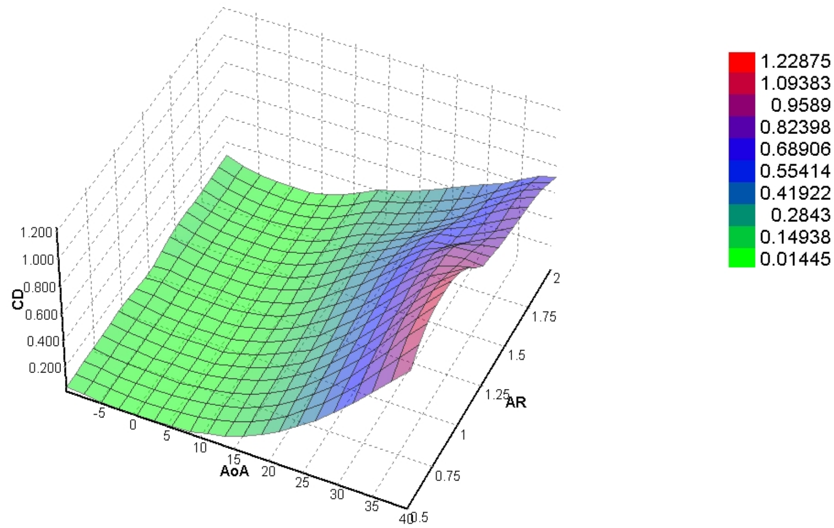


(a) 3D Inverse Zimmerman Re=70K C_L

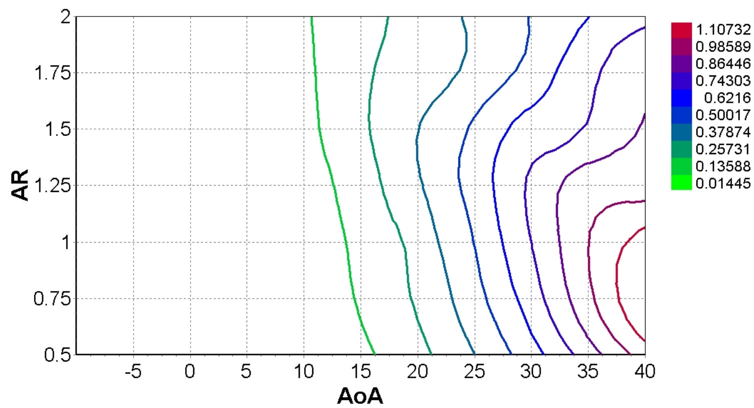


(b) 3D Inverse Zimmerman Re=70K C_L

Figure B.1 Inverse Zimmerman Re=70K C_L

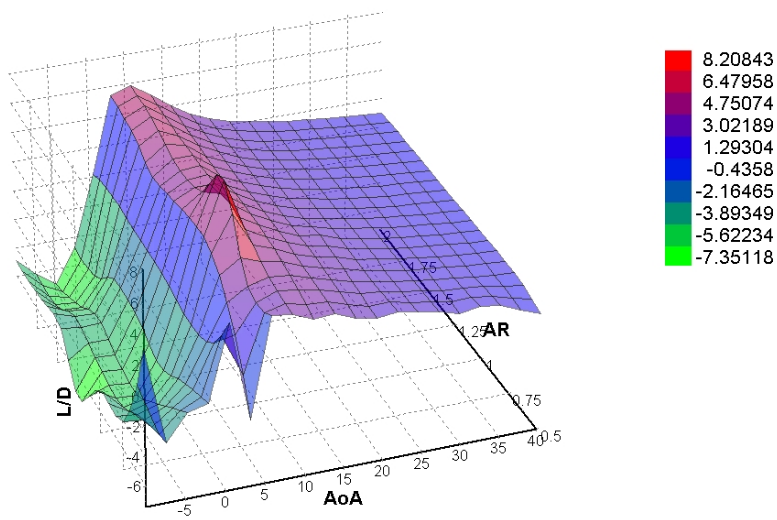


(a) 3D Inverse Zimmerman Re=70K C_D

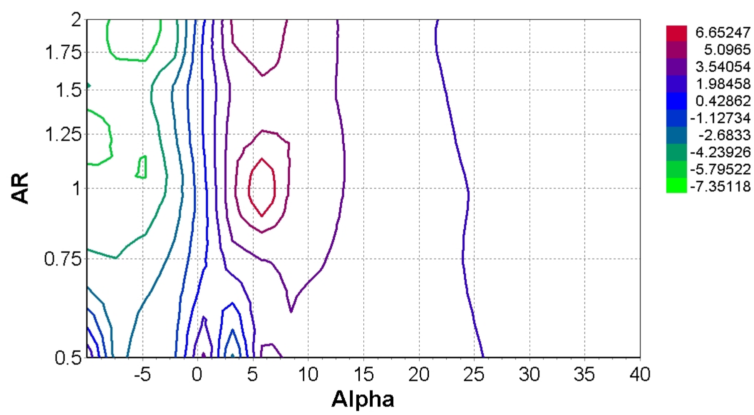


(b) 3D Inverse Zimmerman Re=70K C_D

Figure B.2 Inverse Zimmerman Re=70K C_L



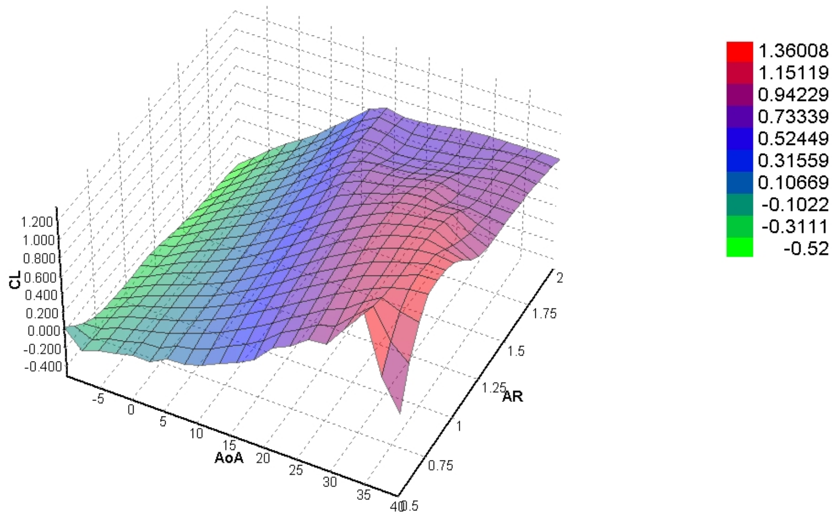
(a) 3D Inverse Zimmerman Re=70K L/D



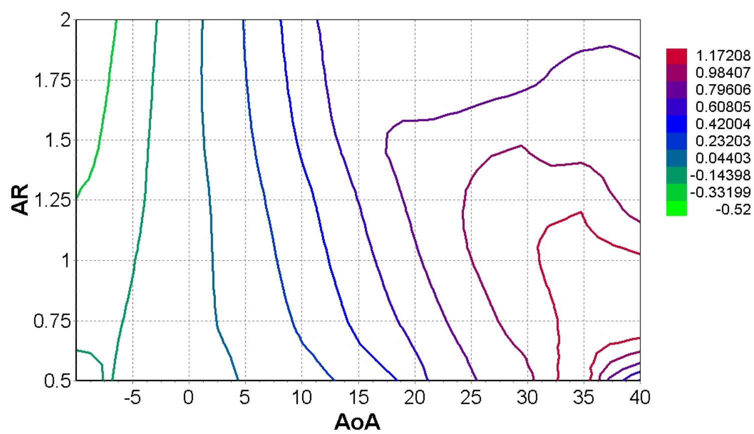
(b) 3D Inverse Zimmerman Re=70K L/D

Figure B.3 Inverse Zimmerman Re=70K L/D

B.2 Inverse Zimmerman at Re=85K (Interpolated)

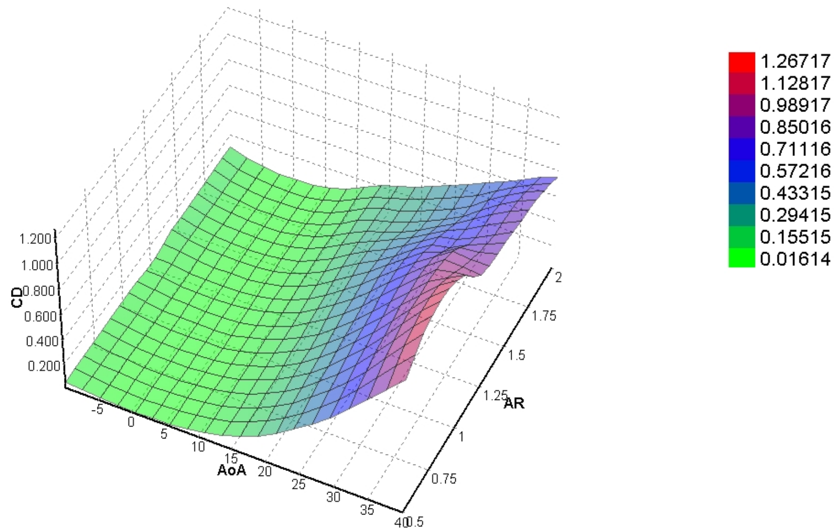


(a) 3D Inverse Zimmerman Re=85K C_L

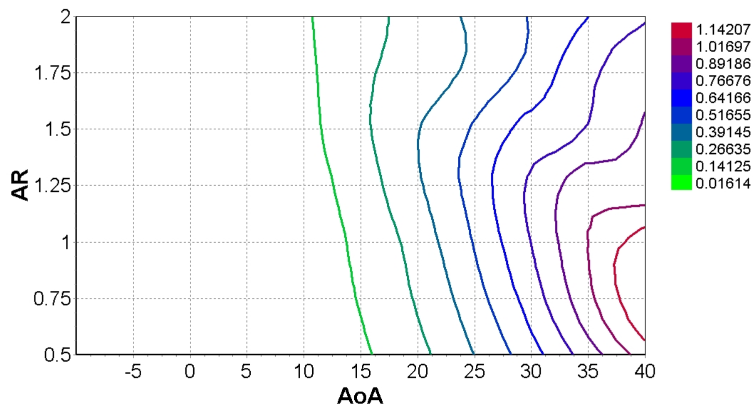


(b) 3D Inverse Zimmerman Re=85K C_L

Figure B.4 Inverse Zimmerman Re=85K C_L

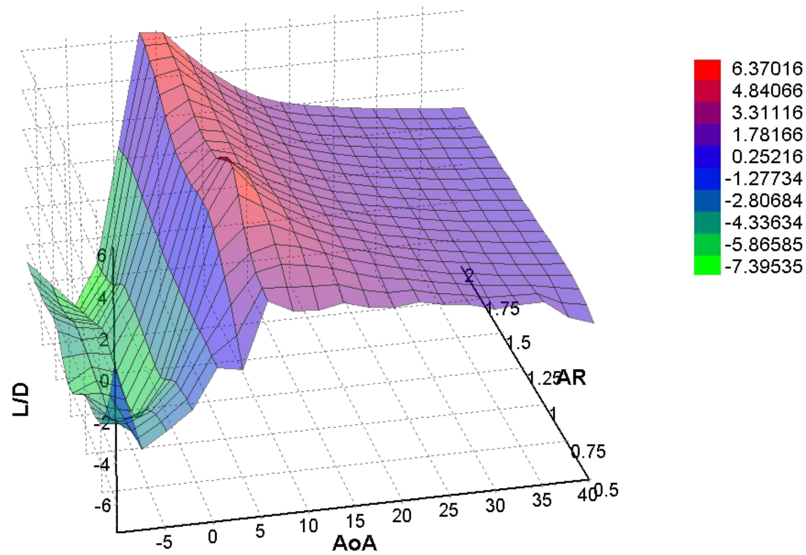


(a) 3D Inverse Zimmerman Re=85K C_D

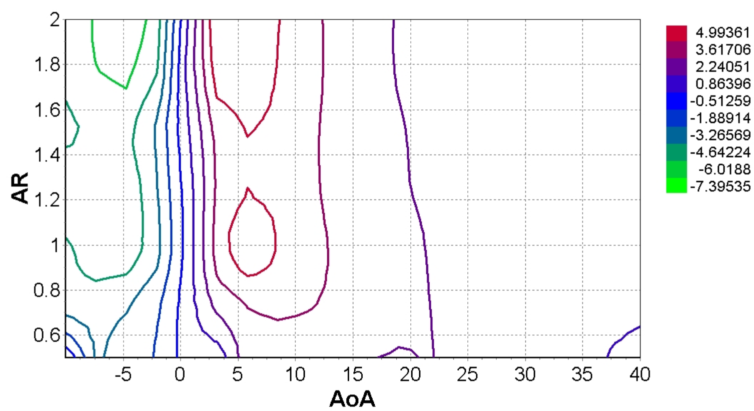


(b) 3D Inverse Zimmerman Re=85K C_D

Figure B.5 Inverse Zimmerman Re=85K C_L



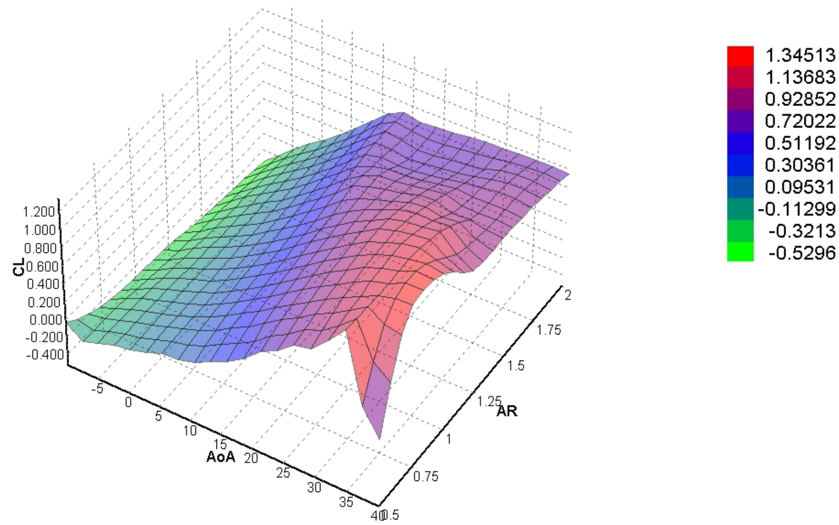
(a) 3D Inverse Zimmerman Re=85K C_D



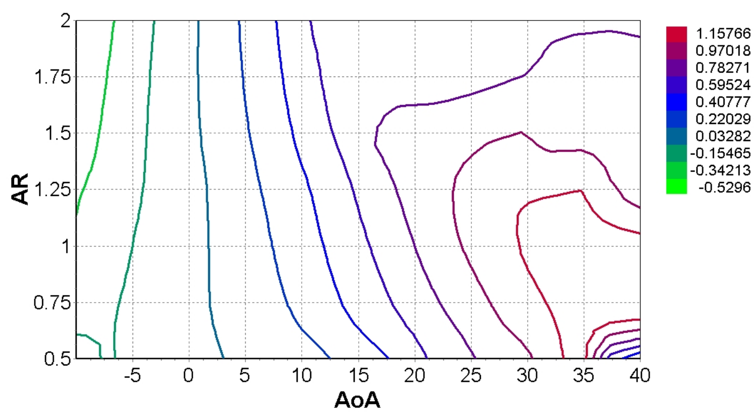
(b) 3D Inverse Zimmerman Re=85K C_D

Figure B.6 Inverse Zimmerman Re=85K L/D

B.3 Inverse Zimmerman at Re=100K [47]

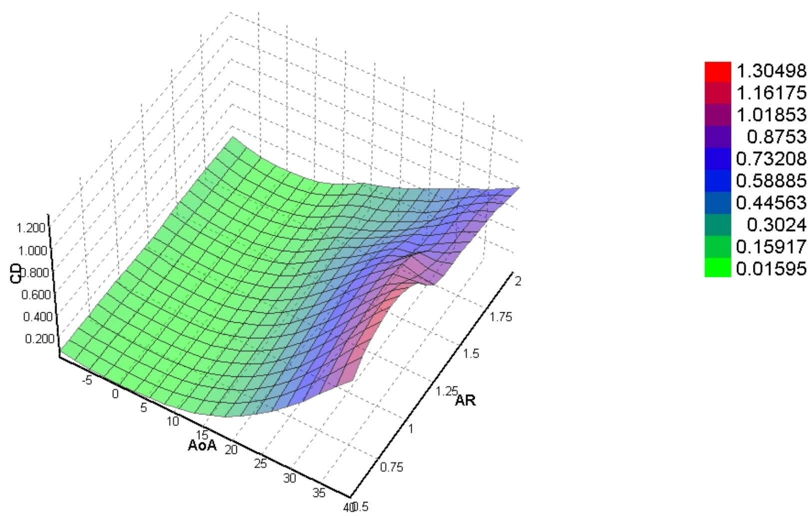


(a) 3D Inverse Zimmerman Re=100K C_L

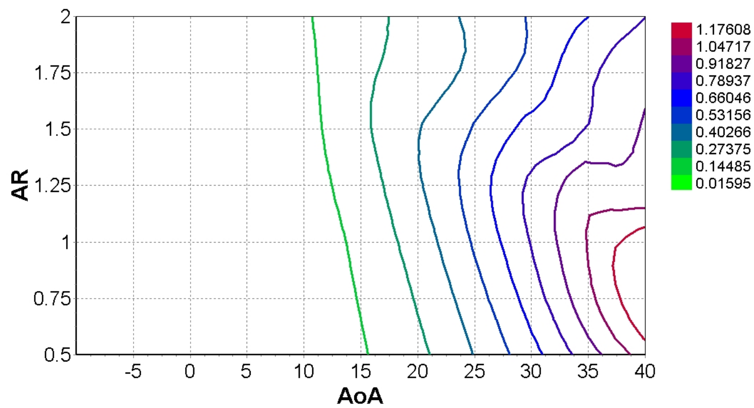


(b) 3D Inverse Zimmerman Re=100K C_L

Figure B.7 Inverse Zimmerman Re=100K C_L

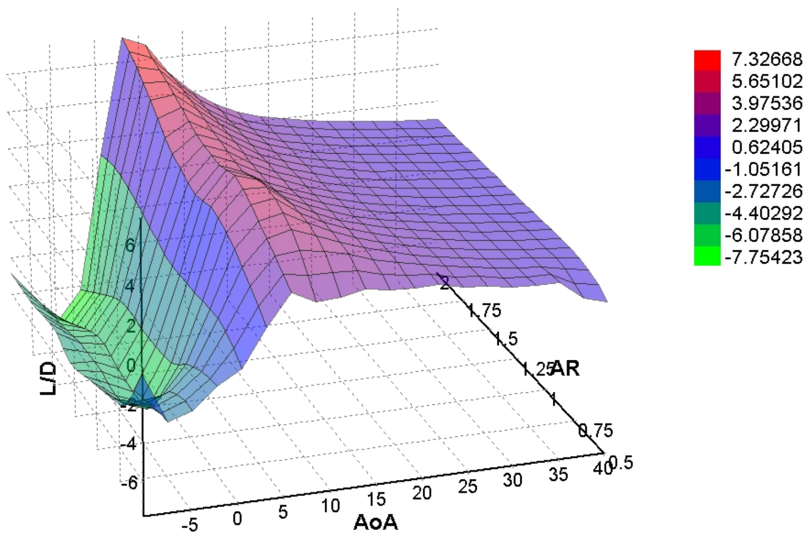


(a) 3D Inverse Zimmerman Re=100K C_D

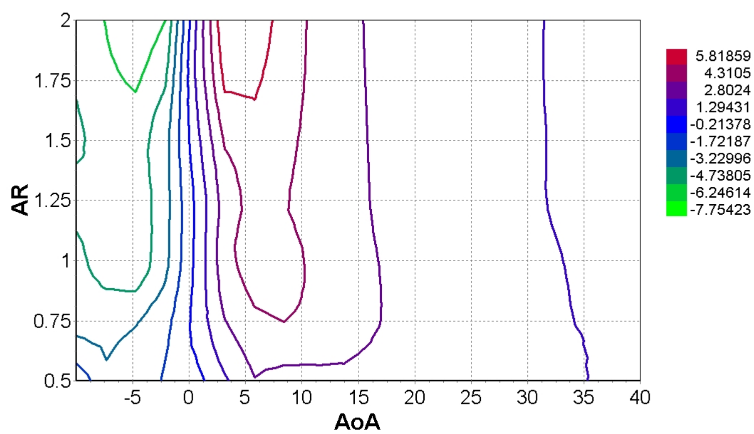


(b) 3D Inverse Zimmerman Re=85K C_D

Figure B.8 Inverse Zimmerman Re=100K C_L



(a) 3D Inverse Zimmerman Re=100K C_D

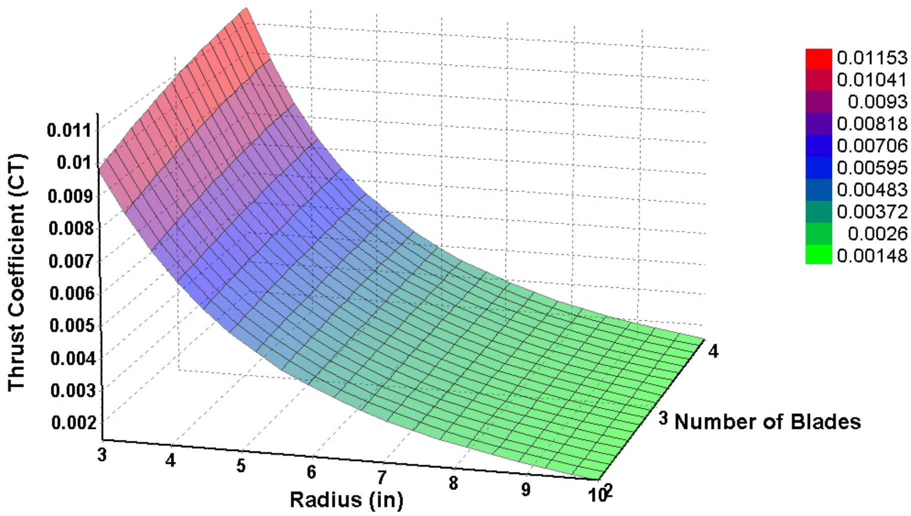


(b) 3D Inverse Zimmerman Re=100K C_D

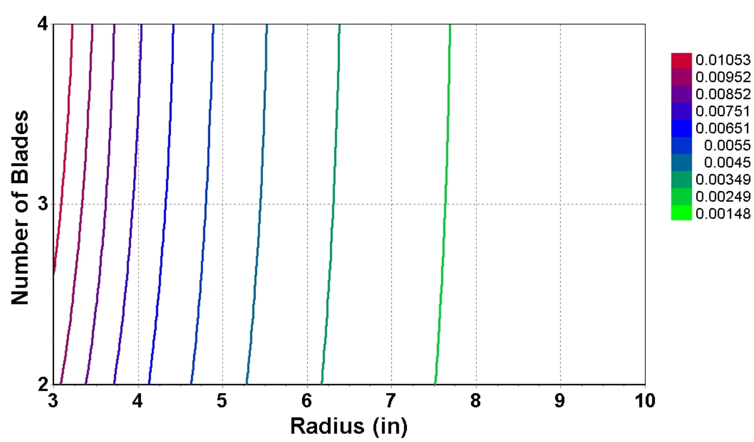
Figure B.9 Inverse Zimmerman Re=100K L/D

Appendix C. QPROP Performance Analysis

C.1 Propeller C_T, I, V and RPM Parameters While Changing Number of Blades and r

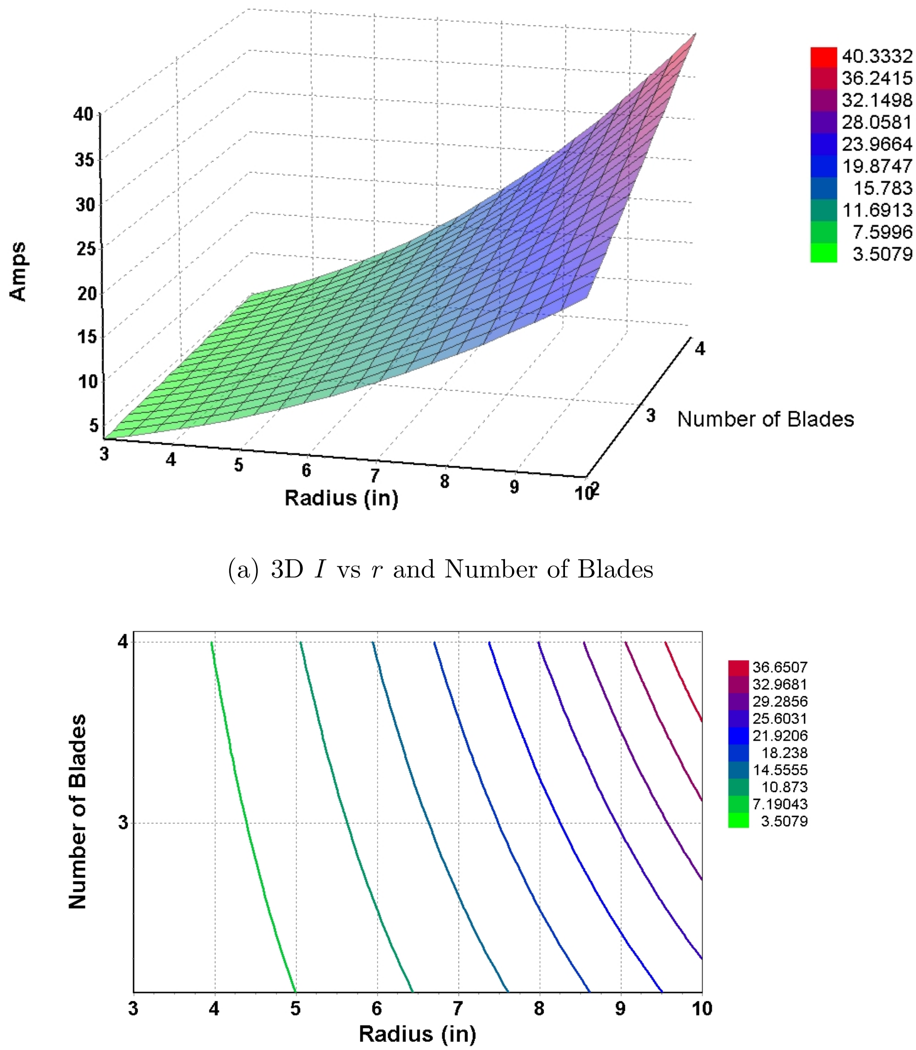


(a) 3D C_T vs r and Number of Blades



(b) 2D C_T vs r and Number of Blades

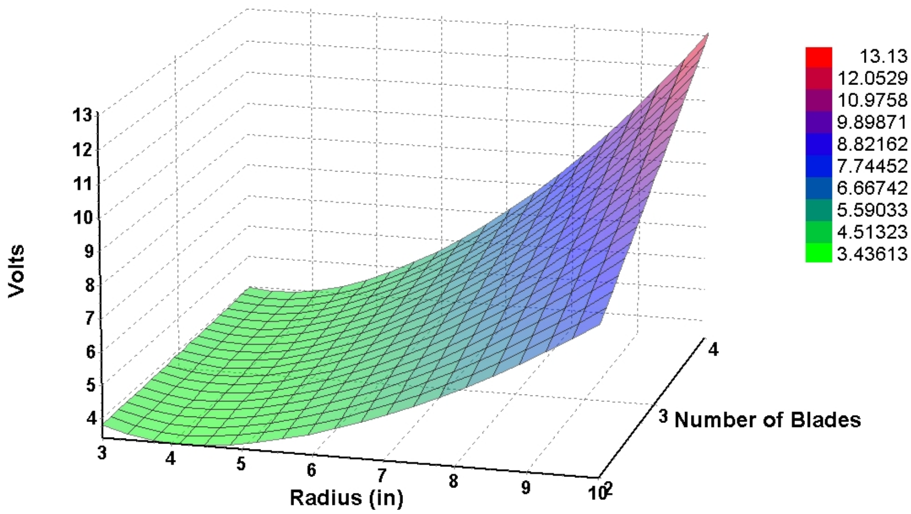
Figure C.1 Change in C_T with r and Number of Blades



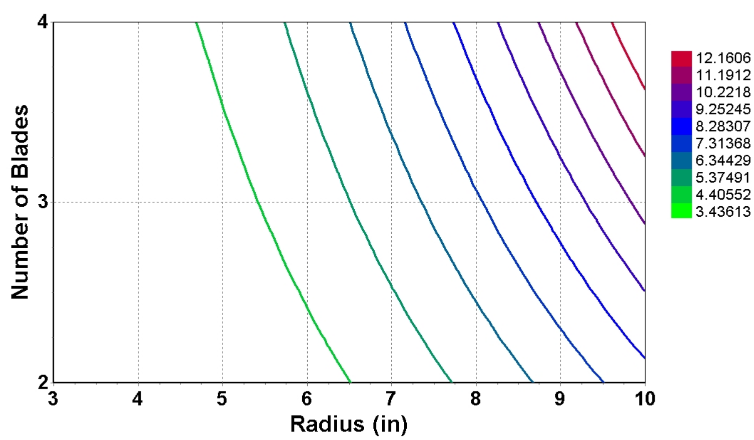
(a) 3D I vs r and Number of Blades

(b) 2D I vs r and Number of Blades

Figure C.2 Change in I with r and Number of Blades

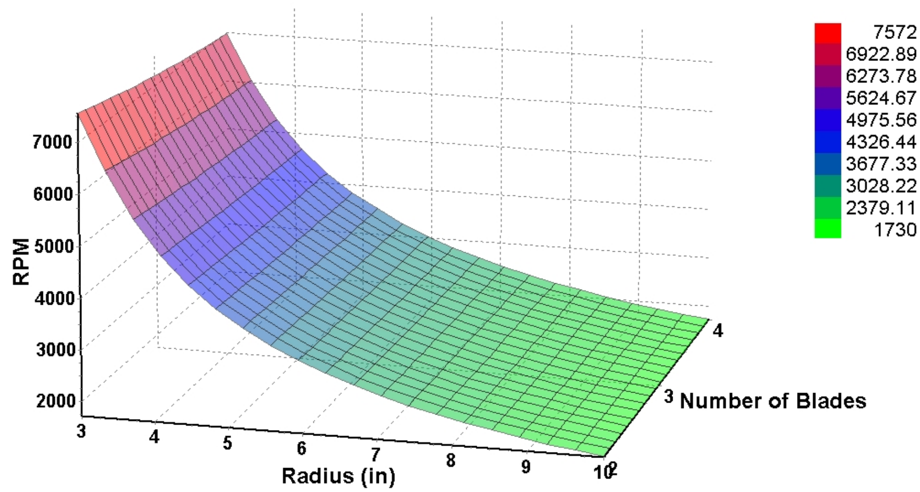


(a) 3D V vs r and Number of Blades

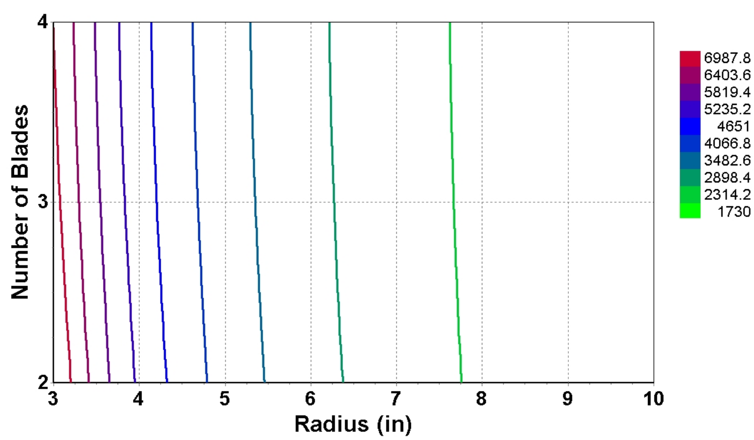


(b) 2D V vs r and Number of Blades

Figure C.3 Change in V with r and Number of Blades

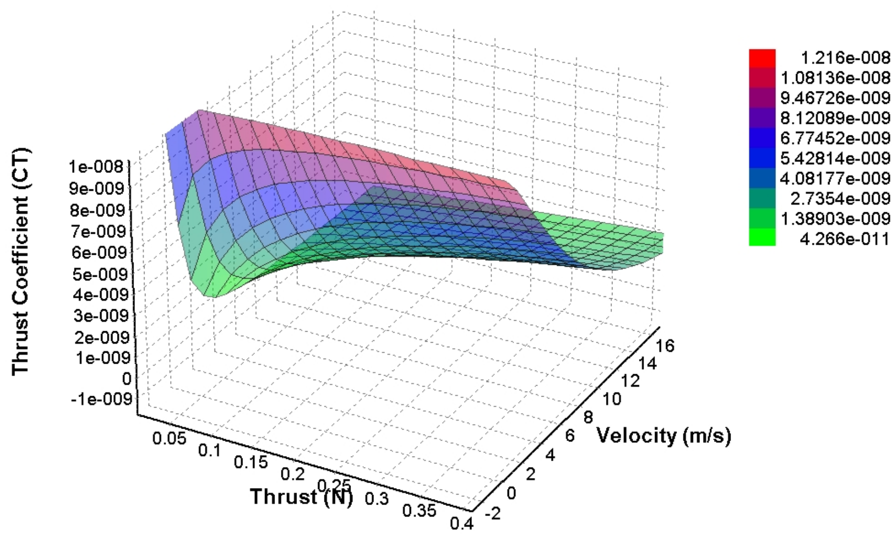


(a) 3D RPM vs r and Number of Blades

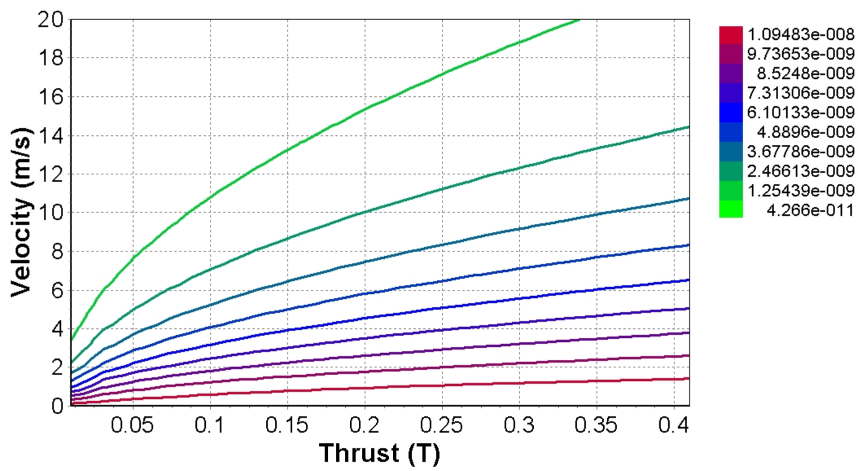


(b) 2D RPM vs r and Number of Blades

Figure C.4 Change in RPM with r and Number of Blades

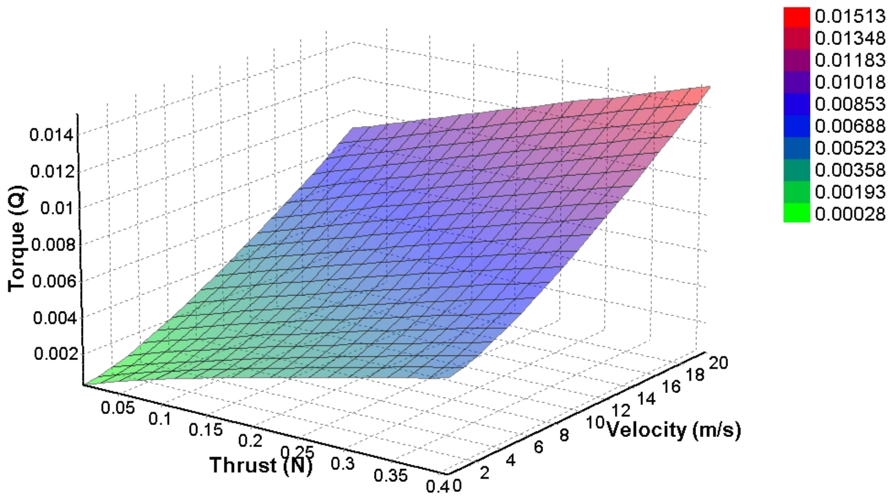


(a) 3D C_T vs T and u

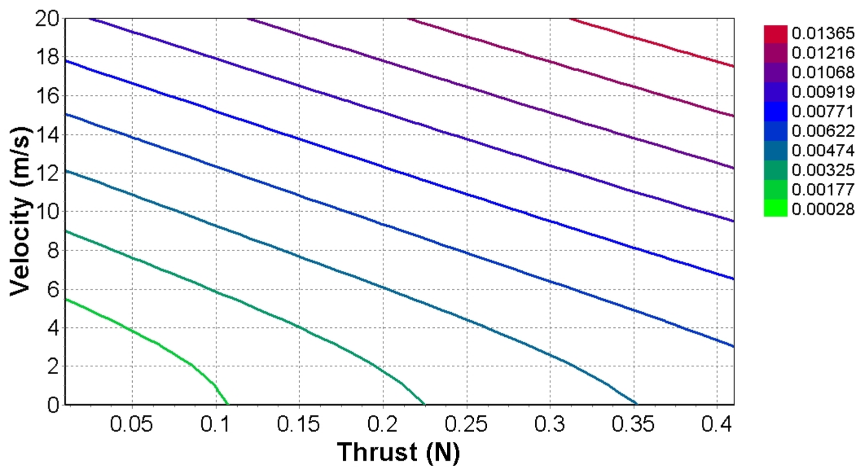


(b) 2D C_T vs T and u

Figure C.5 Change in C_T with T and u

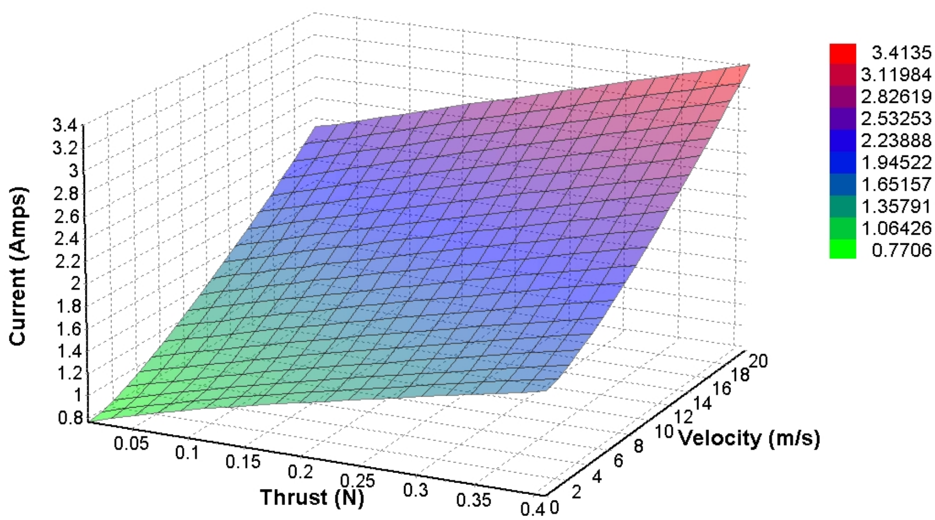


(a) 3D Q vs T and u

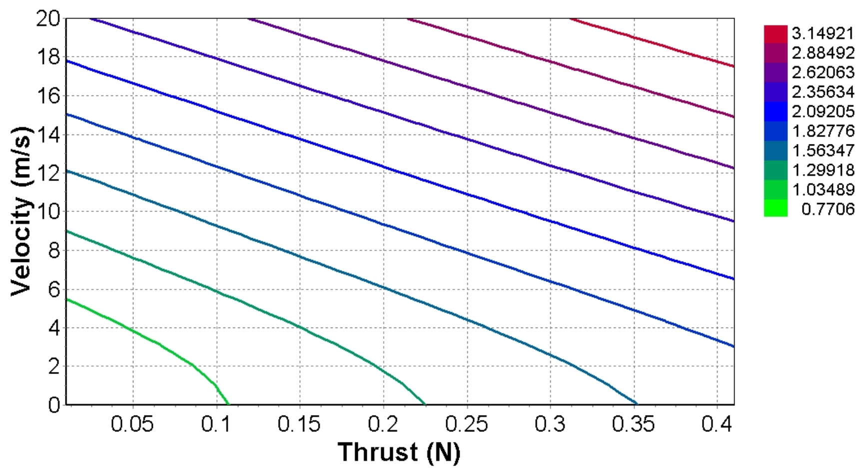


(b) 2D Q vs T and u

Figure C.6 Change in Q with T and u

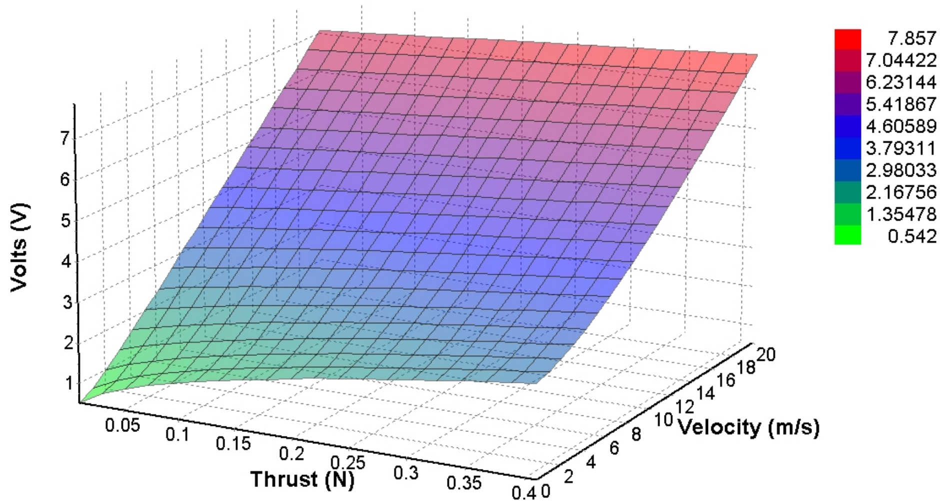


(a) 3D I vs T and u

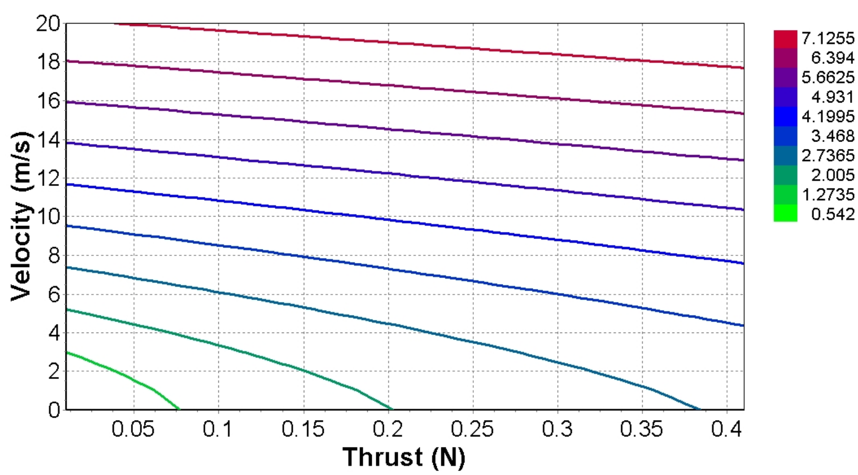


(b) 2D I vs T and u

Figure C.7 Change in I with T and u

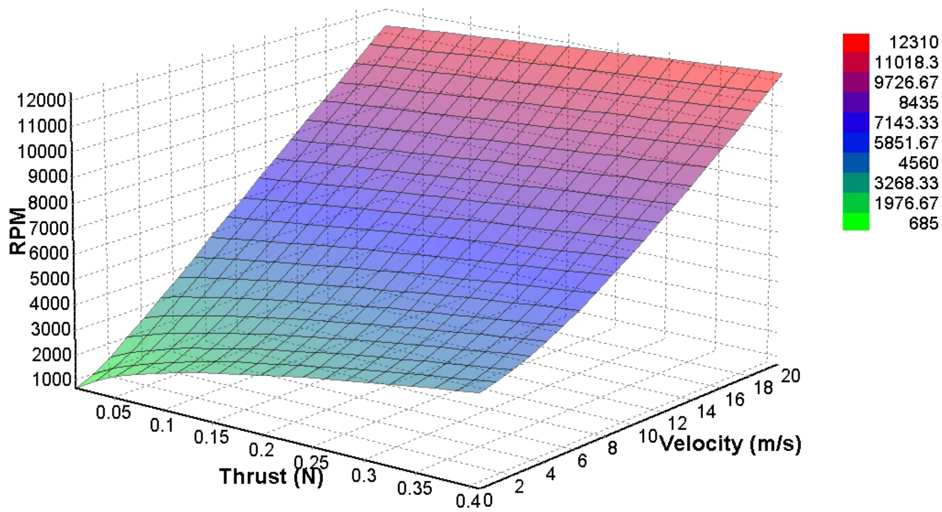


(a) 3D V vs T and u

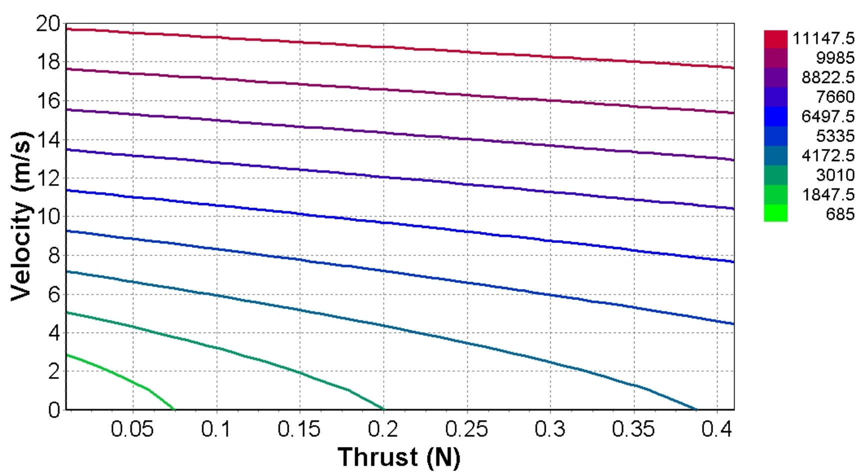


(b) 2D V vs T and u

Figure C.8 Change in V with T and u



(a) 3D RPM vs T and u



(b) 2D RPM vs T and u

Figure C.9 Change in RPM with T and u

Appendix D. Code Used in Integration Process

This appendix includes the code written in MATLAB, MC script files and text files for the components mentioned in Chapter 3.3.

D.1 Geometric Properties

This MATLAB m-file is used in the "Body Definition" component and it creates the geometric variables based on the assumptions that were mentioned in Chapter 3.1.2 on page 3-6.

```
% # variables
% variable: Aspect_ratio double input default="1.66" matlabName="AR"
% variable: Span double input units="inches" matlabName="b"
% variable: Type double input default="3" matlabName="type"
% variable: Sweep_angle_quater_chord_minus4 double output matlabName="sa"
% variable: Root_chord double output  matlabName="rc"
% variable: Tip_Chord double output  matlabName="tc"
% variable: Taper_ratio double output  matlabName="tr"
% variable: sweep_angle_leading_edge_datcom double output matlabName="datcom_sa"
% variable: S_ref_in2 double output   matlabName="S_ref"
% variable: c_bar double output  matlabName="cbar"
% variable: Y_bar double output  matlabName="Ybar"

% minus 4 degrees from sweep angles are just for visualization
%sweep angles are supposed to be quarter chord sweep angles

AR
b                       %inches
type                    %1 rect 2 zim 3 inv zim 4 ell
S_ref=b^2/AR;           %inch^2
%% Rectangular Planform
if type==1
    chord_rect=S_ref/b;               %inches
    taper_ratio_rect=1;
%%these variables are linked to model center
    sweep_agle=0;
    root_chord=chord_rect;
    tip_chord=chord_rect;
    taper_ratio=1;
    datcom_sa=1;
%% Zimmerman Planform
elseif type==2
%for plotting purposes
    S_zim=S_ref;                      %zimmerman planform area
    a_3=b/2;                          %semi-major axis of LEADING edge ellipse
    a_1=S_zim*2/(4*pi*a_3);           %S_zim=pi*(a_3*a_2+3*a_2*a_3)/2   inch^2
    a_2=3*a_1;                        %semi-major axis  of TRAILING edge ellipse
    c_1=a_1*(pi/2-1);                 %equivalent tip chord for LEADING ellipse
    c_2=a_2*(pi/2-1);                 %equivalent tip chord for TRAILING ellipse
    e_1=b/2;                          %Semimajor axis ellipse
    e_2=S_ref/(pi*e_1);               %Semiminor axis ellipse
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
    chord_zim=a_2+a_1;
    t_chord_zim=c_1+c_2;
    taper_ratio_zim=(c_1+c_2)/chord_zim;
    sweep_angle_le_zim=atand(3/4*(a_1-c_1)/a_3)-4;
    sweep_angle_te_zim=atand((a_2-c_2)/a_3);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %%these variables are linked to model center
    sweep_agle=sweep_angle_le_zim;
    root_chord=chord_zim;
    tip_chord=t_chord_zim;
    taper_ratio=taper_ratio_zim;
    datcom_sa=atand((a_1-c_1)/a_3);
%% inverse zimmerman Planform
elseif type==3

    S_zim=S_ref;      %zimmerman planform area
    a_3=b/2;                          %semi-major axis of LEADING edge ellipse
    a_1=S_zim*2/(4*pi*a_3);           %S_zim=pi*(a_3*a_2+3*a_2*a_3)/2   inch^2
    a_2=3*a_1;                        %semi-major axis  of TRAILING edge ellipse

    c_1=a_1*(pi/2-1);                 %equivalent tip chord for LEADING ellipse
    c_2=a_2*(pi/2-1);                 %equivalent tip chord for TRAILING ellipse

    e_1=b/2; %Semimajor axis ellipse
    e_2=S_ref/(pi*e_1);%Semiminor axis ellipse
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    chord_zim=a_2+a_1;
    t_chord_zim=c_1+c_2;
    taper_ratio_zim=(c_1+c_2)/chord_zim;
    sweep_angle_le_zim=atand((a_1-c_1)/a_3);
 % now inverse zimmerman profile parameters
    chord_invzim=a_2+a_1;
    t_chord_invzim=c_1+c_2;
    taper_ratio_invzim=(c_1+c_2)/chord_invzim;
    sweep_angle_le_invzim=atand(3/4*(a_2-c_2)/a_3)-4;
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %%these variables are linked to model center
    sweep_agle=sweep_angle_le_invzim;
    root_chord=chord_invzim;
    tip_chord=t_chord_invzim;
    taper_ratio=taper_ratio_invzim;
    datcom_sa=atand((a_2-c_2)/a_3);
%% Elliptical
elseif type==4
    %for plotting purposes
    S_zim=S_ref;                      %zimmerman planform area
    a_3=b/2;                          %semi-major axis of LEADING edge ellipse
    a_1=S_zim*2/(4*pi*a_3);           %S_zim=pi*(a_3*a_2+3*a_2*a_3)/2   inch^2
    a_2=3*a_1;                        %semi-major axis  of TRAILING edge ellipse
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    c_1=a_1*(pi/2-1);                 %equivalent tip chord for LEADING ellipse
    c_2=a_2*(pi/2-1);                 %equivalent tip chord for TRAILING ellipse
    e_1=b/2;    %Semimajor axis
    e_2=S_ref/(pi*e_1);               %Semiminor axis
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    chord_ellip=2*e_2;
    t_chord_ellip=2*e_2*(pi/2 - 1) ;  %2*(S_ref/(2*e_1)-e_2);
    taper_ratio_ellip=t_chord_ellip/chord_ellip;
    sweep_angle_le_ellip=atand(3/4*(e_2-t_chord_ellip/2)/e_1)-4;
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%these variables are linked to model center
    sweep_agle=sweep_angle_le_ellip;
    root_chord=chord_ellip;
    tip_chord=t_chord_ellip;
    taper_ratio=taper_ratio_ellip;
```

```
    datcom_sa=atand((e_2-t_chord_ellip/2)/e_1);
else disp('please enter integers 1,2,3 or 4 for the type of planform')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Results
sa=sweep_agle;
rc=root_chord;
tc=tip_chord;
tr=taper_ratio;
datcom_sa;
cbar=2/3*rc*(1+tr+tr^2)/(1+tr) %mean aerodynamic chord Raymer p.56
Ybar=b/6*((1+2*tr)/(1+tr))     %''  y location Raymer p.56
```

D.2 Atmospheric Model

This MATLAB m-file is used in the “Air Properties in SI Units” component to find air properties at a given temperature.

```
% # variables
% variable: pressure double output  units="pascall, N/m^2" matlabName="pr"
% variable: speed_of_sound double output  units="m/s" matlabName="a"
% variable: density double output default="1.23" units="kg/m^3" matlabName="rho_r"
% variable: dynamic_viscosity double output  units="Nsec/m^2" matlabName="mu_r"
% variable: kinematic_viscosity double output units="m^2/sec" matlabName="nu_r"
% variable: temperature double input units="celcius" matlabName="t_celc"
%

%altitude model for base
%for altitude <36152 ft

t_celc ;    %base temperature

t_fh=t_celc*1.8+32; %temp in fahrenheit

%alt=alti*3.2808399 ; %altitude in feet  1 meter = 3.2808399 feet
%t_fh=59-0.00356*alt; %temperature in Fahrenheit
%t_celc=(t_fh-32)/1.8; %in celcius  http://www.infoplease.com/ipa/A0001731.html
t_kel=t_celc+273.15; %kelvin degrees

R=287;%gas constant in J/(kg.K)
gamma=1.4; %ratio of specific heats

%from http://www.grc.nasa.gov/WWW/K-12/airplane/atmos.html
%% Properties of Air at Atmospheric Pressure in SI Units
%http://www.engsolcom.com/Database_Pages/Air_Properties.html

%T(C) rho(kg/m^3) mu (N sec/m^2) nu (m^2/sec)

p_si=[  0   1.29  1.72E-05  1.33E-05
        5   1.27  1.74E-05  1.37E-05
        10 1.25 1.77E-05 1.41E-05
        15 1.23 1.79E-05 1.46E-05
        20 1.21 1.81E-05 1.50E-05
        25 1.19 1.84E-05 1.55E-05
        30 1.17 1.86E-05 1.60E-05
        35 1.15 1.88E-05 1.64E-05
        40 1.13 1.91E-05 1.69E-05
```

```
        45 1.11 1.93E-05 1.74E-05
        50 1.09 1.95E-05 1.79E-05
        55 1.08 1.98E-05 1.84E-05
        60 1.06 2.00E-05 1.88E-05
        65 1.04 2.02E-05 1.93E-05
        70 1.03 2.04E-05 1.98E-05
        75 1.01 2.07E-05 2.04E-05
        80 1.00 2.09E-05 2.09E-05
        85 0.99 2.11E-05 2.14E-05
        90 0.97 2.13E-05 2.19E-05
        95 0.96 2.15E-05 2.24E-05
        100 0.95 2.17E-05 2.30E-05];

temp=p_si(:,1);
rho=p_si(:,2);
mu=p_si(:,3);
nu=p_si(:,4);

rho_r=interp1(temp,rho,t_celc,'spline' );
mu_r=interp1(temp, mu,t_celc,'spline' );
nu_r=interp1(temp, nu,t_celc,'spline' );

pr=rho_r*R*t_kel; %pressure in pa
a=sqrt(gamma*R*t_kel); %speed of sound m/sec
```

D.3 Experimental Data Interpolation

"Aero-Block" component utilizes the following background to find the interpolated or extrapolated aerodynamic coefficients.

D.3.1 Data Acquisition from Excel Spreadsheet. Experimental data from an Excel spreadsheet under different tabs and names was converted to xxx.mat files in MATLAB and this process was repeated for all planforms and Re numbers. This section presents the 3D database creation of rectangular and Zimmerman C_D data.

```
%this m file creates database from experimental cl cd cm data


close all;clear all;clc;
alpha=xlsread('cd','rect 70K','A2:A52');
AR=[0.5 0.75 1 1.25 1.50 1.75 2];
Re=[70000;100000];
cd(:,:,1)=xlsread('cd','rect 70K','B2:H52');
cd(:,:,2)=xlsread('cd','rect 100K','B2:H52');
[AR_X,alpha_Y,Re_Z]=meshgrid(AR,alpha,Re);

save cd_rect.mat
%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
close all;clear all;clc;

alpha=xlsread('cd','zim 70K','A2:A52');
AR=[0.5 0.75 1 1.25 1.50 1.75 2];
```

```
Re=[70000;100000];
cd(:,:,1)=xlsread('cd','zim 70K','B2:H52');
cd(:,:,2)=xlsread('cd','zim 100K','B2:H52');
[AR_X,alpha_Y,Re_Z]=meshgrid(AR,alpha,Re);

save cd_zim.mat
```

D.3.2 Interpolation/Extrapolation. This section presents how C_L, C_D *and* C_M data were interpolated/extrapolated.

```
% # variables
% variable: type double input  matlabName="type"
% variable: AR_r double input  matlabName="AR_r"
% variable: Re_r double input  matlabName="Re_r"
% variable: alpha_r double input  matlabName="alpha_r"
% variable: CL double output format="0.000" matlabName="CL"
% variable: CD double output  format="0.000" matlabName="CD"
% variable: CM double output  format="0.000" matlabName="CM"

% subscript r doesn't mean anything, just to seperate variables
% it is critically important to specify xxx.mat locations in ModelCenter

type
AR_r
Re_r
alpha_r

if type==1
    load 'aero_interpolation\Cl_rect.mat'
elseif type==2
    load 'aero_interpolation\Cl_zim.mat'
elseif type==3
    load 'aero_interpolation\Cl_ziminv.mat'
elseif type==4
    load 'aero_interpolation\Cl_ell.mat'
else disp('Please enter integers 1 to 4 for the type of planform ')
end

CLi=interp3(AR_X,alpha_Y,Re_Z,CL,AR_r,alpha_r,Re_r,'spline');

if type==1
    load 'aero_interpolation\cd_rect.mat'
elseif type==2
    load 'aero_interpolation\cd_zim.mat'
elseif type==3
    load 'aero_interpolation\cd_ziminv.mat'
elseif type==4
    load 'aero_interpolation\cd_ell.mat'
else disp('Please enter integers 1 to 4 for the type of planform ')
end

cdi=interp3(AR_X,alpha_Y,Re_Z,cd,AR_r,alpha_r,Re_r,'spline');

if type==1
    load 'aero_interpolation\cm_rect.mat'
elseif type==2
    load 'aero_interpolation\cm_zim.mat'
elseif type==3
    load 'aero_interpolation\cm_ziminv.mat'
elseif type==4
    load 'aero_interpolation\cm_ell.mat'    %location of xxx.mat file
```

```
else disp('Please enter integers 1 to 4 for the type of planform ')
end

cmi=interp3(AR_X,alpha_Y,Re_Z,cm,AR_r,alpha_r,Re_r,'spline');

%% Operation below was needed  due to the sensitivity
%% of DATCOM for number of digits after the decimal point

CL1=sprintf('%0.4f',CLi);
CL=str2num(CL1)

CD1=sprintf('%0.4f',cdi)
CD=str2num(CD1)

CM1=sprintf('%0.6f',cmi)
CM=str2num(CM1)
```

D.4 Determination of Aerodynamic Coefficients

The ModelCenter Selectively Running Switch component is used to run one of the two different DATCOM filewrappers conditionally dependent upon the Reynolds number. Under this component, there are three subcomponents: two different DATCOM filewrappers and a script file. Selectively running the function is accomplished via the MC script file. It is critically important that the user uncheck the "Prevalidate Inputs" under the script component settings (options) for the script file to run selectively and without prevalidating.

```
# variables
variable: Re_r double input
variable: cd1 double input
variable: cd2 double input
variable: cm1 double input
variable: cm2 double input
variable: cl1 double input
variable: cl2 double input
variable: CD double output
variable: CM double output
variable: CL double output
variable: L_D double output

sub run
if Re_r < 170000 then
CL=cl1
CD=cd1
CM=cm1

else
CL=cl2
CD=cd2
CM=cm2

end if
```

```
L_D=abs(CL/CD)
end sub
```

D.5 Flight Parameters and Input Converter

The Re number after linking all the variables is found via this component. It also has a digit converter for DATCOM input files. As experienced, DATCOM would fail if the related numbers were directly linked to DATCOM filewrappers. Another important point to mention here is that DATCOM Re number must be in (1/Length) format.

```
% # variables
% variable: Reynolds_number double output  matlabName="Re_r"
% variable: chord double input matlabName="chord"
% variable: angle_of_attack double input  matlabName="aoa"
% variable: Kinematic_viscosity double input  matlabName="nu_r"
% variable: velocity double input  matlabName="u"
% variable: Mach_number double input  format="0.000" matlabName="Mach"
%
% setGroup "MC_inputs_for_Datcom"
% variable: Mach double input  matlabName="Mach"
% variable: Span double input  matlabName="Span"
% variable: Root_chord double input matlabName="Root_chord"
% variable: Tip_chord double input  matlabName="Tip_chord"
% variable: Sweep double input  matlabName="Sweep"
% variable: Sref double input  matlabName="Sref"
% variable: XCG double input  matlabName="XCG"
% variable: ZCG double input  matlabName="ZCG"
% variable: XW double input  matlabName="XW"
% variable: ZW double input d matlabName="ZW"
% variable: X5_loc double input  matlabName="X5_loc"
%
% setGroup "DATCOM_INPUTS"
% variable: datcom_re double output  matlabName="datcom_re"
% variable: datcom_mach double output  matlabName="datcom_mach"
% variable: datcom_span double output  matlabName="datcom_span"
% variable: datcom_rootchord double output  matlabName="datcom_rootchord"
% variable: datcom_tipchord double output  matlabName="datcom_tipchord"
% variable: datcom_sweep double output  matlabName="datcom_sweep"
% variable: datcom_Sref double output  matlabName="datcom_Sref"
% variable: datcom_semispan double output matlabName="datcom_semispan"
% variable: datcom_xcg double output  matlabName="datcom_XCG"
% variable: datcom_zcg double output  matlabName="datcom_ZCG"
% variable: datcom_XW double output  matlabName="datcom_XW"
% variable: datcom_ZW double output  matlabName="datcom_ZW"
% variable: datcom_bodyX5 double output  matlabName="datcom_bodyX5"

%Reynold number as an output
Re_r=u*chord*0.0254/nu_r

%Here Modelcenter numbers were converted into
%3 digits after point for datcom use
```

```
M=sprintf('%0.3f',Mach);
datcom_mach=str2num(M)

Re1=u/(nu_r*3.2808399)
%1 meter = 3.2808399 feet
Re=sprintf('%0.3f',Re1);
datcom_re=str2num(Re)

s=sprintf('%0.3f',Span);
ss=sprintf('%0.3f',Span/2);
datcom_span=str2num(s);
datcom_semispan=str2num(ss);

rc=sprintf('%0.3f',Root_chord);
datcom_rootchord=str2num(rc);

tc=sprintf('%0.3f',Tip_chord);
datcom_tipchord=str2num(tc);

sw=sprintf('%0.3f',Sweep);
datcom_sweep=str2num(sw);

sref=sprintf('%0.3f',Sref);
datcom_Sref=str2num(sref);

xcg=sprintf('%0.2f',XCG);
datcom_XCG=str2num(xcg);

zcg=sprintf('%0.2f',ZCG);
datcom_ZCG=str2num(zcg);

xW=sprintf('%0.2f',XW);
datcom_XW=str2num(xW);

zW=sprintf('%0.2f',ZW);
datcom_ZW=str2num(zW);

x5=sprintf('%0.2f',X5_loc);
datcom_bodyX5=str2num(x5);
```

D.6 Flight Data Component

This section has the equations for L, D, T_{req} and W_{appx}. The T and u variables are defined for coaxial forward and hovering flight for linking in MC.

```
% # variables
%
% setGroup "Inputs_for_forces"
% variable: rho double input  matlabName="rho"
% variable: vel double input  matlabName="vel"
% variable: Sref double input  matlabName="Sref"
% variable: AR double input  matlabName="AR"
% variable: span double input  matlabName="span"
% variable: L_over_D double input  matlabName="L_over_D"
% variable: Cl double input  matlabName="Cl"
% variable: Cd double input  matlabName="Cd"
%
% setGroup "Flight_data"
```

```
% variable: Lift double output  matlabName="Lift"
% variable: Drag double output  matlabName="Drag"
% variable: Weight_appx double output  matlabName="Weight_appx"
% variable: Weight_act double output  matlabName="Weight_act"
% variable: Weight_ratio double output  matlabName="W_ratio"
% variable: alpha_stall double output  matlabName="alpha_stall"
% variable: T_req_in_N double output  matlabName="T_req_in_N"
%
% setGroup "Forward_Flight_data"
% variable: Horizontal_velocity double input  matlabName="h_vel"
% variable: Thrust_ratio_f double input  matlabName="Thrust_ratio_f"
% variable: T_req_forward double output  matlabName="T_req_forward"
% variable: T1_f double output  matlabName="T1_f"
% variable: T2_f double output  matlabName="T2_f"
%
% setGroup "Hover_data"
% variable: Vertical_velocity double input matlabName="V_vert"
% variable: Thrust_ratio_h double input  matlabName="Thrust_ratio_h"
% variable: T_req_hover double  matlabName="T_req_hover"
% variable: T1_h double output  matlabName="T1_h"
% variable: T2_h double output  matlabName="T2_h"

g=9.807;
Lift=Cl*(0.5*rho*vel^2*Sref*0.0254^2)/g*1000;% in gr
Drag=Cd*(0.5*rho*vel^2*Sref*0.0254^2)/g*1000;% in gr


alpha_stall=-10*atan(4*(AR-1.25))+28;
Weight_appx=6.8953*span - 19.262; %gr
Weight_act=Weight_appx*1.271;
W_ratio=Weight_act/Weight_appx;

T_req_in_N=Weight_appx/(abs(L_over_D))*g/1000;
T_req_hover=Weight_appx*g/1000;
T_req_forward=T_req_in_N;

%Forward Flight COAXIAL
%T2/T1; %T1 forward, T2 aft; Thrust_ratio max is 1
T1_f=T_req_forward/(1+Thrust_ratio_f);
T2_f=T1_f*Thrust_ratio_f;


%Hover Flight COAXIAL
%T2/T1; %T1 forward, T2 aft; Thrust_ratio max is 1
T1_h=T_req_hover/(1+Thrust_ratio_h);
T2_h=T1_h*Thrust_ratio_h;
```

D.7 DATCOM

Figure 3.4 on page 3-7 and Figure 3.13 on page 3-18 summarize the DATCOM operation and filewrapper structure. This sections presents the related file structures in DATCOM operations.

D.7.1 DATCOM Input File. The DATCOM input file with C_D *and* C_M experimental data supplement for MAV configuration is presented in this section.

$EXPR01 section was changed in the MC Switch script file based on the assumptions mentioned in the related chapter. In this specific case, DATCOM will generate its own data when $EXPR01 section is removed.

```
$FLTCON NMACH=1.0, MACH(1)=0.0296, RNNUB(1)=216170.212437379,
           NALPHA=1.0, ALSCHD(1)=10.0,
 TR=1.0$
 $OPTINS SREF=144.167, BLREF=14.8,
 $SYNTHS XCG=4.0, ZCG=0.55,
          XW=3.0,  ZW=0.5,  ALIW=0.0,
          SCALE=1.0$
 $BODY NX=5.0,
        X(1)=0.0, X(2)=1.5, X(3)=3.0, X(4)=5.0, X(5)=16,
        R(1)=0.0, R(2)=0.25, R(3)=0.5, R(4)=0.5, R(5)=0.5,
        BNOSE=1.0,
        ITYPE=1.0, METHOD=1.0$
 $WGPLNF CHRDR=12.4, CHRDTP=7.07, SSPN=7.4,  SSPNE=7.4,
          SAVSI=28.335,
          CHSTAT=0.0, TWISTA=0.0,
          DHDADI=0.0, TYPE=1.0$
 $WGSCHR NPTS=30.0,TYPEIN=1.0,
   XCORD(1)=0.0,
   7.444E-04, 2.955E-03, 6.565E-03, 1.146E-02, 1.750E-02, 2.450E-02,
   3.224E-02, 4.049E-02, 4.900E-02, 5.000E-02, 1.500E-01, 2.500E-01,
   3.500E-01, 4.500E-01, 5.500E-01, 6.500E-01, 7.500E-01, 8.500E-01,
   9.500E-01, 9.510E-01, 9.595E-01, 9.678E-01, 9.755E-01, 9.825E-01,
   9.885E-01, 9.934E-01, 9.970E-01, 9.993E-01, 1.0,

   YUPPER(1)=0.0,
   1.702E-03, 3.352E-03, 4.900E-03, 6.299E-03, 7.507E-03, 8.487E-03,
   9.209E-03, 9.651E-03, 9.800E-03, 9.800E-03, 9.800E-03, 9.800E-03,
   9.800E-03, 9.800E-03, 9.800E-03, 9.800E-03, 9.800E-03, 9.800E-03,
   9.800E-03, 9.800E-03, 9.651E-03, 9.209E-03, 8.487E-03, 7.507E-03,
   6.299E-03, 4.900E-03, 3.352E-03, 1.702E-03, 0.0,

   YLOWER(1)=0.0,
   -1.7018E-03, -3.3518E-03, -4.9000E-03, -6.2993E-03, -7.5072E-03,
   -8.4870E-03, -9.2090E-03, -9.6511E-03, -9.8000E-03, -9.8000E-03,
   -9.8000E-03, -9.8000E-03, -9.8000E-03, -9.8000E-03, -9.8000E-03,
   -9.8000E-03, -9.8000E-03, -9.8000E-03, -9.8000E-03, -9.8000E-03,
   -9.6511E-03, -9.2090E-03, -8.4870E-03, -7.5072E-03, -6.2993E-03,
   -4.9000E-03, -3.3518E-03, -1.7018E-03, 0.0, $
DIM IN
DERIV DEG
 $EXPR01
    CDWB(1)=0.1152,
    CMWB(1)=0.010031,
 $
CASEID  CD-CM EXPERIMENTAL DATA FOR WING-BODY
NEXT CASE
```

D.7.2 DATCOM Output File. This section has an example of the DATCOM output file supplemented with C_D *and* C_M.

```
1                              AUTOMATED STABILITY AND CONTROL METHODS PER APRIL 1976 VERSION OF DATCOM
```

USER DEFINED WING SECTION

UPPER ABSCISSA	UPPER ORDINATE	LOWER ABSCISSA	LOWER ORDINATE	X-FRACTION CHORD	MEAN LINE	THICKNESS
0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
0.00074	0.00170	0.00074	-0.00170	0.00074	0.00000	0.00340
0.00296	0.00335	0.00295	-0.00335	0.00296	0.00000	0.00670
0.00657	0.00490	0.00656	-0.00490	0.00657	0.00000	0.00980
0.01146	0.00630	0.01146	-0.00630	0.01146	0.00000	0.01260
0.01750	0.00751	0.01750	-0.00751	0.01750	0.00000	0.01501
0.02450	0.00849	0.02450	-0.00849	0.02450	0.00000	0.01697
0.03224	0.00921	0.03224	-0.00921	0.03224	0.00000	0.01842
0.04049	0.00965	0.04049	-0.00965	0.04049	0.00000	0.01930
0.04900	0.00980	0.04900	-0.00980	0.04900	0.00000	0.01960
0.05000	0.00980	0.05000	-0.00980	0.05000	0.00000	0.01960
0.15000	0.00980	0.15000	-0.00980	0.15000	0.00000	0.01960
0.25000	0.00980	0.25000	-0.00980	0.25000	0.00000	0.01960
0.35000	0.00980	0.35000	-0.00980	0.35000	0.00000	0.01960
0.45000	0.00980	0.45000	-0.00980	0.45000	0.00000	0.01960
0.55000	0.00980	0.55000	-0.00980	0.55000	0.00000	0.01960
0.65000	0.00980	0.65000	-0.00980	0.65000	0.00000	0.01960
0.75000	0.00980	0.75000	-0.00980	0.75000	0.00000	0.01960
0.85000	0.00980	0.85000	-0.00980	0.85000	0.00000	0.01960
0.95000	0.00980	0.95000	-0.00980	0.95000	0.00000	0.01960
0.95100	0.00980	0.95100	-0.00980	0.95100	0.00000	0.01960
0.95950	0.00965	0.95950	-0.00965	0.95950	0.00000	0.01930
0.96780	0.00921	0.96780	-0.00921	0.96780	0.00000	0.01842
0.97550	0.00849	0.97550	-0.00849	0.97550	0.00000	0.01697
0.98250	0.00751	0.98250	-0.00751	0.98250	0.00000	0.01501
0.98850	0.00630	0.98850	-0.00630	0.98850	0.00000	0.01260
0.99340	0.00490	0.99340	-0.00490	0.99340	0.00000	0.00980
0.99700	0.00335	0.99700	-0.00335	0.99700	0.00000	0.00670
0.99930	0.00170	0.99930	-0.00170	0.99930	0.00000	0.00340
1.00000	0.00000	1.00000	0.00000	1.00000	0.00000	0.00000

1 AUTOMATED STABILITY AND CONTROL METHODS PER APRIL 1976 VERSION OF DATCOM

WING SECTION DEFINITION

0 IDEAL ANGLE OF ATTACK = 0.00001 DEG.

ZERO LIFT ANGLE OF ATTACK = 0.00004 DEG.

IDEAL LIFT COEFFICIENT = 0.00000

ZERO LIFT PITCHING MOMENT COEFFICIENT = -0.00001

MACH ZERO LIFT-CURVE-SLOPE = 0.10115 /DEG.

LEADING EDGE RADIUS = 0.00232 FRACTION CHORD

MAXIMUM AIRFOIL THICKNESS = 0.01960 FRACTION CHORD

DELTA-Y = 0.75341 PERCENT CHORD

0**** REYNOLDS NUMBER TOO LOW FOR THE AIRFOIL SECTION MODULE, SECTION CHARACTERISTICS BASED ON A VALUE OF 2.718E5 ***

0 MACH= 0.0296 LIFT-CURVE-SLOPE = 0.08218 /DEG. XAC = 0.25573

1 AUTOMATED STABILITY AND CONTROL METHODS PER APRIL 1976 VERSION OF DATCOM

CHARACTERISTICS AT ANGLE OF ATTACK AND IN SIDESLIP

WING-BODY CONFIGURATION

CD-CM EXPERIMENTAL DATA FOR WING-BODY

FLIGHT CONDITIONS						REFERENCE DIMENSIONS				
MACH NUMBER	ALTITUDE	VELOCITY	PRESSURE	TEMPERATURE	REYNOLDS NUMBER	REF. AREA	REFERENCE LENGTH LONG.	MOMENT REF. CENTER LAT.	HORIZ	VERT
	IN	IN/SEC	LB/IN**2	DEG R	1/FT	IN**2	IN	IN	IN	IN
0 0.030					2.1617E+05	144.167	9.978	14.800	4.000	0.550

0 -------------------DERIVATIVE (PER DEGREE)-------------------

0 ALPHA CD CL CM CN CA XCP CLA CMA CYB CNB CLB

```
0
  10.0   0.115   0.417   0.0100  0.430   0.041   0.023    0.000E+00    0.000E+00    -Infinity   8.832E-06   -2.689E-03
0*NOTE* OUTPUT REFLECTS EXPERIMENTAL DATA INPUTS
1          THE FOLLOWING IS A LIST OF ALL INPUT CARDS FOR THIS CASE.
0
1 END OF JOB.
```

D.7.3 DATCOM Filewrapper. This section has an example of the DATCOM filewrapper structure.

```
#
# Basic Digital DATCOM File Wrapper
#
# @author: Mustafa Turan
# @version: 6 Nov 2008
# @description: MAV DIGITAL DATCOM WRAPPER
#

RunCommands
{
# Put ModelCenter values in the input file
 generate inputFile
# Run the code
 run "digdat"
# Parse the standard output file
 parse outputfile
}

RowFieldInputFile inputFile
{
 templateFile:         mtrn_mav_CL_only.template
 fileToGenerate:  for005.dat

# These are the variables that are being modified in the template file to create the DATCOM input file.
# Other variables can be added as desired/needed.

 setDelimiters "= ,"

 setGroup Flt_Cond
markAsBeginning "$FLTCON"
keyvar: Mach_number            double      "MACH(1)"         description="mach number evaluated"
keyvar: Reynolds_number         double      "RNNUB(1)"        description="reynolds number normalized"
keyvar: AOA                 double      "ALSCHD(1)"       description="angle of attack"

 setGroup optins
markAsBeginning "$OPTINS"
keyvar: S_ref   double  "SREF"    description="reference area"
keyvar: Span      double  "BLREF"  description="span"

 setGroup Components
markAsBeginning "$SYNTHS"
#        MC Var Name  Var Type Code Variable Description Field
keyvar: WingApex_X         double      "XW"             description="Wing Apex location from nose in X dir"
keyvar: WingApex_Z         double      "ZW"             description="Wing Apex location z dir"
keyvar: CG_X               double      "XCG"            description="Center of gravity in the X dir"
keyvar: CG_Z            double     "ZCG"         description="Center of gravity in the Z dir"

 setGroup Fuselage
markAsBeginning "$BODY"
keyvar: FusPos2        double      "X(2)"          description="Rear x-pos of nose cone"
```

```
keyvar: FusPos3        double      "X(3)"          description="Mid x-pos of midsection"
keyvar: FusPos4        double      "X(4)"          description="Rear x-pos of midsection"
keyvar: FusPos5        double      "X(5)"          description="Rear x-pos of AftSection"
keyvar: FusRad2        double      "R(2)"          description="Rear Rad of nose cone"
keyvar: FusRad3        double      "R(3)"          description="Mid rad of midsection"
keyvar: FusRad4        double      "R(4)"          description="Rear rad of midsection"
keyvar: FusRad5        double      "R(5)"          description="Rear rad of Aftsection"

 setGroup Wing
markAsBeginning "$WGPLNF"
keyvar: RootChord        double      "CHRDR"          description="Wing root chord length"
keyvar: TipChord         double      "CHRDTP"         description="Wing tip chord length"
keyvar: SemiSpan         double      "SSPN"           description="Wing semi-span length"
keyvar: Exposed_Semispan double      "SSPNE"          description="Exposed Wing semi-span length"
keyvar: Sweep            double      "SAVSI"          description="Sweep Angle (variable sweep inboard)"

 setGroup experimental_data_input
markAsBeginning "$EXPR01"
keyvar: CD_wing        double      "CDWB(1)"          description="experimental cd input"
# keyvar: CL_wing      double      "CLWB(1)"          description="experimental cl input"
keyvar: CM_wing        double      "CMWB(1)"          description="experimental cm input"

}

RowFieldOutputFile outputFile
{
# This routine parses the program output file.
# Other variables can be extracted as desired.

 fileToParse: for006.dat

 setDelimiters "= "

# Search reference string as before
 markAsBeginning "FLIGHT CONDITIONS"
 setGroup Input_check
 variable: Mach_no           double   5      2
 variable: Re                double   5     3
 variable: Ref_Area             double   5      4
 variable: Ref_Chord            double   5      5
 variable: Ref_Span             double   5      6

 setGroup Coefficients

 variable: CD        double  9 2
 variable: CL      double  9 3
 variable: CM      double 9 4

}
```

D.8 QPROP

Figure 3.10 on page 3-12 and Figure 3.13 on page 3-18 summarize the QPROP operation and filewrapper structure. This sections presents the related file structures in QPROP operations.

D.8.1 QPROP Input Files. There are four different input files in the MCDA tool for QPROP: three of them are the actual input files and the fourth one is the QPROP batch file that runs the QPROP component.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
qcon.def File Template
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

1.225    ! rho  kg/m^3
1.78E-5  ! mu   kg/m-s
340.0    ! a    m/s

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Propeller File Template
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

 2           ! Nblades

 0.50  5.8   ! CL0     CL_a
 -0.3  1.2   ! CLmin   CLmax

 0.028  0.050  0.050  0.5   !  CD0    CD2u  CD2l    CLCD0
 70000   -0.7        !  REref  REexp


 0.0254  0.0254   1.0  !  Rfac   Cfac   Bfac
 0.0     0.0      4.0  !  Radd   Cadd   Badd

#  r    chord    beta
 0.75    0.66    27.5
 1.00    0.69    22.0
 1.50    0.63    15.2
 2.00    0.55    10.2
 2.50    0.44     6.5
 2.875   0.30     4.6
 3.00    0.19     4.2

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Motor File Template
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

 mtrn_motor file

 1     ! motor type (brushed DC)

 0.18     ! Rmotor (Ohms)
 0.72     ! Io     (Amps)
 1700.0   ! Kv     (rpm/Volt)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
QPROP Batch File Template for Velocity and Thrust Inputs
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

qprop mtrn_prop mtrn_motor 3.0 0.0 0.0  0.0 0.981 0.0 0.0 0.0  >  mtrn_Qprop.dat
```

D.8.2 QPROP Output File. There is a single output file generated by the QPROP batch file. Line numbers in the presented example in this section may not match with QPROP filewrapper in the next section.

```
# QPROP Version 1.22
#
# mtrn prop Graupner CAM 6x3 folder xxx
#
#  mtrn_motor file
#   0.31000      Rmotor (Ohms)
#   0.77000      Io     (Amps)
#    2760.0      Kv     (rpm/Volt)
#
#   rho =  1.2250     kg/m^3
#   mu  = 0.17800E-04 kg/m-s
#   a   =  340.00     m/s
#
   1         2        3          4          5          6            7         8        9          10         11
#  V(m/s)    rpm      Dbeta      T(N)       Q(N-m)     Pshaft(W)    Volts     Amps     effmot     effprop    adv

   0.000    1787.      0.000    2.000      0.5720E-01    10.70       6.011   17.3033    0.1029    0.0000    0.00000

   12         13         14         15       16           17          18         19
#  CT         CP         DV(m/s)    eff      Pelec        Pprop       cl_avg     cd_avg

 0.5502E-01  0.1033E-01   6.6896   0.0000   104.0        0.000       0.7406  0.5524E-01#


#  radius   chord   beta     Cl       Cd        Re    Mach    effi     effp    Wa(m/s)     Aswirl      adv_wake
   0.0404  0.0340  30.380   1.1161  0.13131    16121  0.020   0.0000   0.7511   2.834       24.30       0.1196
   0.0450  0.0347  28.311   1.0628  0.11116    18524  0.023   0.0000   0.7653   3.001       22.75       0.1237
   0.0495  0.0350  26.471   1.0176  0.09678    20807  0.025   0.0000   0.7745   3.144       21.36       0.1271
   0.0541  0.0350  24.856   0.9816  0.08656    22902  0.028   0.0000   0.7798   3.268       20.10       0.1299
   0.0587  0.0347  23.442   0.9536  0.07920    24781  0.031   0.0000   0.7824   3.375       18.97       0.1323
   0.0632  0.0341  22.191   0.9315  0.07374    26446  0.033   0.0000   0.7829   3.467       17.93       0.1343
   0.0678  0.0334  21.065   0.9132  0.06952    27920  0.036   0.0000   0.7819   3.546       16.99       0.1360
   0.0724  0.0327  20.026   0.8963  0.06605    29244  0.038   0.0000   0.7798   3.613       16.12       0.1373
   0.0770  0.0319  19.037   0.8786  0.06294    30477  0.041   0.0000   0.7770   3.667       15.30       0.1382
   0.0815  0.0312  18.071   0.8583  0.05996    31670  0.043   0.0000   0.7735   3.708       14.54       0.1388
   0.0861  0.0305  17.130   0.8361  0.05719    32808  0.046   0.0000   0.7692   3.737       13.82       0.1390
   0.0907  0.0298  16.219   0.8130  0.05469    33863  0.049   0.0000   0.7641   3.756       13.14       0.1389
   0.0952  0.0291  15.344   0.7901  0.05253    34796  0.051   0.0000   0.7578   3.764       12.49       0.1384
   0.0998  0.0283  14.511   0.7681  0.05075    35564  0.054   0.0000   0.7503   3.761       11.87       0.1377
   0.1044  0.0274  13.726   0.7479  0.04938    36118  0.056   0.0000   0.7415   3.750       11.29       0.1367
   0.1090  0.0264  12.988   0.7297  0.04838    36446  0.059   0.0000   0.7315   3.731       10.73       0.1355
   0.1135  0.0254  12.296   0.7130  0.04768    36573  0.062   0.0000   0.7204   3.705       10.20       0.1341
   0.1181  0.0244  11.647   0.6975  0.04722    36532  0.064   0.0000   0.7082   3.675       9.710       0.1326
   0.1227  0.0233  11.039   0.6825  0.04692    36361  0.067   0.0000   0.6953   3.643       9.252       0.1311
   0.1273  0.0223  10.469   0.6671  0.04672    36108  0.069   0.0000   0.6818   3.614       8.834       0.1298
   0.1318  0.0213   9.937   0.6509  0.04666    35723  0.072   0.0000   0.6677   3.591       8.463       0.1287
   0.1364  0.0200   9.449   0.6357  0.04718    34787  0.074   0.0000   0.6510   3.572       8.127       0.1278
   0.1410  0.0182   9.014   0.6219  0.04892    32745  0.077   0.0000   0.6292   3.559       7.827       0.1272
   0.1455  0.0156   8.638   0.6055  0.05295    28970  0.079   0.0000   0.5975   3.577       7.615       0.1277
   0.1501  0.0119   8.329   0.5448  0.06159    22816  0.082   0.0000   0.5425   3.827       7.905       0.1368
```

D.8.3 QPROP Filewrapper. This section has an example of the QPROP filewrapper structure.

```
#
# Basic QPROP filewrapper
#
# @author: Mustafa Turan
# @version: 7 Nov 2008
# @description: Mav QPROP analysis
#

RunCommands
{
# Put ModelCenter values in the input file
 generate inputFile1

 generate inputFile2

 generate inputFile3

 generate inputfile4

# Run the code
 run "qprop_batch.bat"
# Parse the standard output file
 parse outputfile
}

RowFieldInputFile inputFile1
{
templateFile:  qcon.template
fileToGenerate:        qcon.def

setDelimiters "= ,"
setGroup Flight_conditions
variable: rho double 1 1  description="density"
variable: mu double 2 1  description="dynamic viscosity"
variable: a double 3 1  description="speed of sound"
}

RowFieldInputFile inputFile2

{
templateFile:  qprop_batch.template
fileToGenerate:  qprop_batch.bat

setDelimiters "= ,"

variable: Velocity  double 1 4  default=1.0 description="flight velocity"
variable: RPM   double 1 5  default=0.0 description="RPM"
variable: Volt    double 1 6  default=0 description="Volt"
variable: dBeta   double 1 7  default=0.0 description="dBeta"
variable: Thrust    double 1 8  default=1.0 description="Thrust required"
variable: Torque  double 1 9  default=0 description="Torque"
variable: Amps    double 1 10 default=0 description="Amps"
variable: Pele  double 1 11 default=0 description="Power "
}

RowFieldInputFile inputFile3

 {
 templateFile:  mtrn_motor.template
```

```
 fileToGenerate:        mtrn_motor

 setDelimiters " ,"
 setGroup inputs_Motor
 variable: Rmotor double 6 1  description="Rmotor(ohms)"
 variable: Io double 7 1    description="Io (amps)"
 variable: Kv double 8 1   description="Kv (rpm/volts)"
 }

RowFieldInputFile inputFile4

 {
 templateFile:  mtrn_prop.template
 fileToGenerate:        mtrn_prop

 setDelimiters " ,"
 setGroup inputs_Prop
 variable: Blade_Number double 4 1  description="Number of blades"

 setGroup radius
 variable: r1 double 17 1    description="radius 1 "
 variable: r2 double 18 1     description="radius 2 "
 variable: r3 double 19 1     description="radius 3 "
 variable: r4 double 20 1     description="radius 4 "
 variable: r5 double 21 1   description="radius 5 "
 variable: r6 double 22 1   description="radius 6 "
 variable: r7 double 23 1   description="radius 7 "

 setGroup chord
 variable: c1 double 17 2    description="chord 1 "
 variable: c2 double 18 2     description="chord 2 "
 variable: c3 double 19 2     description="chord 3 "
 variable: c4 double 20 2     description="chord 4 "
 variable: c5 double 21 2   description="chord 5 "
 variable: c6 double 22 2   description="chord 6 "
 variable: c7 double 23 2   description="chord 7 "

 setGroup beta
 variable: b1 double 17 3    description="beta 1 "
 variable: b2 double 18 3     description="beta 2 "
 variable: b3 double 19 3     description="beta 3 "
 variable: b4 double 20 3     description="beta 4 "
 variable: b5 double 21 3   description="beta 5 "
 variable: b6 double 22 3   description="beta 6 "
 variable: b7 double 23 3   description="beta 7 "
 }

RowFieldOutputFile outputFile

{
# This routine parses the program output file.
# Other variables can be extracted as desired.

 fileToParse: mtrn_Qprop.dat

 setDelimiters "= "

 markAsBeginning "V(m/s)"
 setGroup prop_erties
 variable: Velocity double 2 2
 variable: RPM double 2 3
 variable: Dbeta double 2 4
 variable: Thrust double 2 5
 variable: Q      double 2  6
 variable: Pshaft double 2 7
```

```
 variable: Volts double 2 8
 variable: Amps double 2  9
 variable: effmot double 2  10
 variable: effprop double 2  11
 variable: adv double 2  12
 variable: CT double 2  13
 variable: CP double 2  14
 variable: DV double 2  15
 variable: eff double 2  16
 variable: Pelec double 2  17
 variable: Pprop double 2  18
 variable: cl_avg double 2  19
 variable: cd_avg double 2  20
}
```

D.9 AVL

Figure 3.12 on page 3-15 and Figure 3.13 on page 3-18 summarize the AVL operation and filewrapper structure. This section presents the related file structures in AVL operations. The geometry input file is based on the Figure 3.25 on page 3-36.

D.9.1 AVL Input Files. There are four different input files in the MCDA tool for AVL: three of them are the actual input files and the fourth one is the auxiliary AVL batch file that changes parameters in the AVL menu.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Geometry Input File Template (xxx.avl)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

MAV Mustafa Turan
#Mach
 0.0
#IYsym   IZsym   Zsym
 0       0       0.0
#Sref    Cref    Bref
 32.0     4.0995     8.0
#Xref    Yref    Zref
1.5     0.0     0.0
#
#
#=====================================================================
SURFACE
Wing
#Nchordwise  Cspace   Nspanwise   Sspace
8            1.0       12         1.0
#
YDUPLICATE
0.0
#
ANGLE
0.0
#--------------------------------------------------------------
SECTION
```

```
#Xle    Yle    Zle     Chord   Ainc  Nspanwise  Sspace
0.      0.     0.      5.093     0.0   0          0

AFILE
mtrn_MAV_wing.dat

#Cname   Cgain  Xhinge  HingeVec     SgnDup
CONTROL
flap     1.0   0.75    0.0 0.0 0.0   1.0
#-------------------------------------------------------------added by mtrn
SECTION
#Xle    Yle    Zle     Chord   Ainc  Nspanwise  Sspace
0.5      2.5     0.      3.8     0.0   0          0

AFILE
mtrn_MAV_wing.dat

#Cname   Cgain  Xhinge  HingeVec     SgnDup
CONTROL
flap     1.0   0.75    0.0 0.0 0.0   1.0

CONTROL
aileron  -1.0   0.75    0.0 0.0 0.0  -1.0

CLAF
1.0
#-------------------------------------------------------------
SECTION
#Xle    Yle    Zle     Chord   Ainc  Nspanwise  Sspace
0.9     4.0    0.0     2.097     0.0   0          0

AFILE
mtrn_MAV_wing.dat

CONTROL
aileron  -1.0   0.75    0.0 0.0 0.0  -1.0
#
CLAF
1.0
#====================================================================
SURFACE
H-stab
#Nchordwise  Cspace   Nspanwise   Sspace
 6           1.0      6           1.0
#
YDUPLICATE
0.0
#
TRANSLATE
7.0 0.0 0.0
#
#-------------------------------------------------------------
SECTION
#Xle    Yle    Zle     Chord   Ainc  Nspanwise  Sspace
0.0     0.0    0.0     1.5     0.    0          0

#Cname   Cgain  Xhinge  HingeVec     SgnDup
CONTROL
elevator 1.0    0.7     0.0 1.0 0.0  1.0
#-------------------------------------------------------------
SECTION
#Xle    Yle    Zle     Chord   Ainc  Nspanwise  Sspace
0.14    2.0   0.0     1.0    0.    0          0

#Cname   Cgain  Xhinge  HingeVec     SgnDup
```

```
CONTROL
elevator 1.0    0.7     0.0 1.0 0.0  1.0
#
#=====================================================================
SURFACE
V-stab
#Nchordwise  Cspace    Nspanwise  Sspace
6            1.0       5          1.0
TRANSLATE
7.0 0.0 0.0
#--------------------------------------------------------------
SECTION
#Xle    Yle    Zle     Chord   Ainc  Nspanwise  Sspace
0.0     0.     0.0     1.4     0.    0          0

#Cname   Cgain  Xhinge  HingeVec     SgnDup
CONTROL
rudder   1.0    0.5     0.0 0.0 1.0  1.0
#--------------------------------------------------------------
SECTION
#Xle    Yle    Zle     Chord   Ainc  Nspanwise  Sspace
0.14    0.     1.5     0.8    0.    0          0

#Cname   Cgain  Xhinge  HingeVec     SgnDup
CONTROL
rudder   1.0    0.5     0.0 0.0 1.0  1.0
#--------------------------------------------------------------

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Mass File Example from the Manual(xxx.mass)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

#  SuperGee
#
#  Dimensional unit and parameter data.
#  Mass & Inertia breakdown.

#  Names and scalings for units to be used for trim and eigenmode calculations.
#  The Lunit and Munit values scale the mass, xyz, and inertia table data below.
#  Lunit value will also scale all lengths and areas in the AVL input file.
Lunit = 0.0254 m
Munit = 0.001  kg
Tunit = 1.0    s

#-------------------------
#  Gravity and density to be used as default values in trim setup (saves runtime typing).
#  Must be in the unit names given above (i.e. m,kg,s).
g   = 9.81
rho = 1.225

#-------------------------
#  Mass & Inertia breakdown.
#  x y z  is location of item's own CG.
#  Ixx... are item's inertias about item's own CG.
#
#  x,y,z system here must be exactly the same one used in the .avl input file
#     (same orientation, same origin location, same length units)
#
#  mass      x     y     z         Ixx     Iyy    Izz    [ Ixy   Ixz   Iyz ]
*   1.       1.    1.    1.        1.       1.      1.       1.    1.    1.
+   0.       0.    0.    0.        0.       0.      0.       0.    0.    0.
   58.0     3.34  12.0   1.05   4400      180     4580               ! right wing
   58.0     3.34 -12.0   1.05   4400      180     4580               ! left wing
   16.0    -5.2    0.0   0.0       0       80       80               ! fuselage pod
   18.0    13.25   0.0   0.0       0      700      700               ! boom+rods
```

```
   22.0    -7.4    0.0   0.0       0        0        0               ! battery
    2.0    -2.5    0.0   0.0       0        0        0               ! jack
    9.0    -3.8    0.0   0.0       0        0        0               ! RX
    9.0    -5.1    0.0   0.0       0        0        0               ! rud servo
    6.0    -5.9    0.0   0.0       0        0        0               ! ele servo
    9.0     2.6    1.0   0.0       0        0        0               ! R wing servo
    9.0     2.6   -1.0   0.0       0        0        0               ! L wing servo
    2.0     1.0    0.0   0.5       0        0        0               ! wing connector
    1.0     3.0    0.0   0.0       0        0        0               ! wing pins
    6.0    29.0    0.0   1.0      70        2       72               ! stab
    6.0    33.0    0.0   2.0      35       39        4               ! rudder
    0.0    -8.3    0.0   0.0       0        0        0               ! nose wt.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Run File Example from the Manual(xxx.run) Up to Five Different Run Scenerion
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Run case  1:   -unnamed-

 alpha        ->  CL          =   1.12345
 beta         ->  beta        =   0.00000
 pb/2V        ->  pb/2V       =   0.00000
 qc/2V        ->  qc/2V       =   0.00000
 rb/2V        ->  rb/2V       =   0.00000
 flap         ->  flap        =   0.00000
 aileron      ->  Cl roll mom =   0.00000
 elevator     ->  Cm pitchmom =   0.00000
 rudder       ->  Cn yaw  mom =   0.00000

 alpha     =   8.24035
 beta      =   0.00000
 pb/2V     = -0.645134E-24
 qc/2V     =   0.00000
 rb/2V     = -0.834026E-25
 CL        =   1.12345
 CDo       =   0.00000
 bank      =   20.00000
 elevation =   0.00000
 heading   =   0.00000
 Mach      =   0.00000
 velocity  =   5.8633
 density   =  0.200000
 grav.acc. =   10.0000
 turn_rad. =   0.00000
 load_fac. =   1.00000
 X_cg      =  0.650000
 Y_cg      =   0.00000
 Z_cg      =   0.00000
 mass      =   100.000
 Ixx       =   400.000
 Iyy       =   200.000
 Izz       =   600.000
 Ixy       =   0.00000
 Iyz       =   0.00000
 Izx       =   0.00000
 visc CL_a =   0.00000
 visc CL_u =   0.00000
 visc CM_a =   0.00000
 visc CM_u =   0.00000

 ---------------------------------------------
 Run case  2:   -unnamed-

 alpha        ->  CL          =   0.87654
 beta         ->  beta        =   0.00000
 pb/2V        ->  pb/2V       =   0.00000
 qc/2V        ->  qc/2V       =   0.00000
```

```
 rb/2V        ->  rb/2V       =   0.00000
 flap         ->  flap        =   0.00000
 aileron      ->  Cl roll mom =   0.00000
 elevator     ->  Cm pitchmom =   0.00000
 rudder       ->  Cn yaw  mom =   0.00000

 alpha     =   2.34567
 beta      =   0.00000
 pb/2V     =   0.0000
 qc/2V     =   0.74000E-03
 rb/2V     =  -0.134026E-01
 CL        =   0.789102
 CDo       =   0.01234
 bank      =   20.00000
 elevation =   0.00000
 heading   =   0.00000
 Mach      =   0.00000
 velocity  =   5.8633
 density   =  0.200000
 grav.acc. =   10.0000
 turn_rad. =   0.00000
 load_fac. =   1.00000
 X_cg      =  0.650000
 Y_cg      =   0.00000
 Z_cg      =   0.00000
 mass      =   100.000
 Ixx       =   400.000
 Iyy       =   200.000
 Izz       =   600.000
 Ixy       =   0.00000
 Iyz       =   0.00000
 Izx       =   0.00000
 visc CL_a =   0.00000
 visc CL_u =   0.00000
 visc CM_a =   0.00000
 visc CM_u =   0.00000

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Auxiliary Batch File Template for AVL Menu Manuplation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

OPER
A R 0.5555E-01
A P 0.5526E-03
A Y 0.1687E-02
A D1 0.
A D2 0.
A D3 0.
A D4 0.
A A 1.5
X
ST
mtrn_MAV_results.txt
O

QUIT
```

D.9.2 AVL Output File. AVL generates output after running the following batch file. As seen in the command, xxx.run file Auxiliary Batch file is called within the command while operating.

AVL main batch file (as executable in the filewrapper):

```
avl mtrn_MAV.avl vanilla.run < mtrn_MAV_b.batch
```

```
---------------------------------------------------------------
Vortex Lattice Output -- Total Forces

Configuration: MAV Mustafa Turan
    # Surfaces =   5
    # Strips   =  41
    # Vortices = 294

 Sref =  20.515       Cref =  4.4230       Bref =  5.9055
 Xref = 0.65000       Yref =  0.0000       Zref =  0.0000

Standard axis orientation,  X fwd, Z down

Run case: -unnamed-

 Alpha =   6.00000     pb/2V =   0.05507     p'b/2V =   0.05555
 Beta  =   0.00000     qc/2V =   0.00055
 Mach  =     0.000     rb/2V =   0.00748     r'b/2V =   0.00169

 CXtot =  -0.00282     Cltot =   0.00000     Cl'tot =   0.00000
 CYtot =   0.00550     Cmtot =   0.00000
 CZtot =  -0.28209     Cntot =   0.00000     Cn'tot =   0.00000

 CLtot =   0.28025
 CDtot =   0.03229
 CDvis =   0.00000     CDind =   0.03229
 CLff  =   0.28313     CDff  =  -0.37659    | Trefftz
 CYff  =  -0.00652         e =   -0.0399    | Plane

  flap           =   0.00000
  aileron        =   3.22334
  elevator       = -27.34651
  rudder         =  17.87431

---------------------------------------------------------------

Stability-axis derivatives...

                             alpha                beta
                  ----------------    ----------------
z' force CL |    CLa =   2.317367    CLb =  -0.003174
y  force CY |    CYa =   0.097712    CYb =  -0.024071
x' mom.  Cl'|    Cla =  -0.005580    Clb =  -0.117160
y  mom.  Cm |    Cma =  -0.533778    Cmb =   0.004271
z' mom.  Cn'|    Cna =  -0.043476    Cnb =   0.033274

                     roll rate  p'     pitch rate  q'       yaw rate  r'
                  ----------------    ----------------    ----------------
z' force CL |    CLp =   0.001552    CLq =   3.537171    CLr =   0.004538
y  force CY |    CYp =   0.216254    CYq =   0.093204    CYr =   0.026703
x' mom.  Cl'|    Clp =  -0.158943    Clq =   0.005837    Clr =   0.149331
```

```
 y  mom.  Cm |     Cmp =  -0.002085    Cmq =  -1.652294    Cmr =  -0.011797
 z' mom.  Cn'|     Cnp =  -0.084672    Cnq =  -0.044089    Cnr =  -0.048904

                  flap        d1     aileron      d2     elevator     d3     rudder       d4
                 ----------------    ----------------    ----------------    ----------------
 z' force CL |   CLd1 =   0.011449   CLd2 =   0.000110   CLd3 =   0.003959   CLd4 =   0.000024
 y  force CY |   CYd1 =   0.000296   CYd2 =   0.000111   CYd3 =   0.000042   CYd4 =  -0.000384
 x' mom.  Cl'|   Cld1 =   0.000026   Cld2 =   0.002533   Cld3 =   0.000006   Cld4 =  -0.000006
 y  mom.  Cm |   Cmd1 =  -0.004345   Cmd2 =  -0.000062   Cmd3 =  -0.004591   Cmd4 =  -0.000034
 z' mom.  Cn'|   Cnd1 =  -0.000133   Cnd2 =  -0.000145   Cnd3 =  -0.000033   Cnd4 =   0.000360
 Trefftz drag| CDffd1 =  -0.026130 CDffd2 =   0.000054 CDffd3 =   0.015024 CDffd4 =   0.000072
 span eff.   |    ed1 =  -0.000498    ed2 =  -0.000034    ed3 =  -0.002704    ed4 =  -0.000010



 Neutral point  Xnp =   1.668789

 Clb Cnr / Clr Cnb  =   1.153100    (  > 1 if spirally stable )
```

D.9.3 AVL Filewrapper. This section has an example of the AVL filewrapper structure.

```
#
# Athena Vortex Lattice filewrapper
#
# @author: Mustafa Turan
# @version: 23 Jan 2009
# @description: MAV AVL analysis
#

RunCommands
{
# Put ModelCenter values in the input file
 generate inputFile1

 generate inputFile2

# Run the code
 run "avl_batch.bat"
# Parse the standard output file
 parse outputfile
}

RowFieldInputFile inputFile1
{
templateFile:  mtrn_MAV.template
fileToGenerate:        mtrn_MAV.avl

setDelimiters " ,"
setGroup "UserInputs.Geometry_input_file"
variable: Mach    double 3 1  description="keep it zero for M<0.2"
variable: S_ref double 7 1  description="reference ares"
variable: C_ref double 7 2  description="c_bar"
variable: b_ref double 7 3  description="span"
variable: X_ref double 9 1  description="see manual"
variable: Y_ref double 9 2  description="see manual"
variable: Z_ref double 9 3  description="see manual"


setGroup "UserInputs.Wing.Section_1"
variable: Xle1 double 26 1  description="see manual"
variable: Yle1 double 26 2  description="see manual"
```

```
variable: Zle1 double 26 3  description="see manual"
variable: Chord double 26 4  description="see manual"
variable: Flap_Cgain double 33 2  description="see manual"
variable: Flap_Xhinge  double 33 3 description="see manual"
variable: Flap_SgnDup  double 33 7 description="see manual"

setGroup "UserInputs.Wing.Section_2"
variable: Xle2 double 37 1  description="see manual"
variable: Yle2 double 37 2  description="see manual"
variable: Zle2 double 37 3  description="see manual
variable: Chord double 37 4  description="see manual"
variable: Flap_Cgain double 44 2 description="see manual"
variable: Flap_Xhinge  double 44 3 description="see manual"
variable: Flap_SgnDup double 44 7  description="see manual"
variable: Aileron_Cgain double 47 2  description="see manual"
variable: Aileron_Xhinge  double 47 3  description="see manual"
variable: Aileron_SgnDup double 47 7  description="see manual"

setGroup "UserInputs.Wing.Section_3"
variable: Xle3 double 54 1  description="see manual"
variable: Yle3 double 54 2  description="see manual"
variable: Zle3 double 54 3  description="see manual"
variable: Chor double 54 4  description="see manual"
variable: Aileron_Cgain double 60 2  description="see manual"
variable: Aileron_Xhinge  double 60 3  description="see manual"
variable: Aileron_SgnDup double 60 7 description="see manual"

setGroup "UserInputs.Horizontal_STAB.Translate"
variable: Translate_x double 74 1 description="see manual"
variable: Translate_y double 74 2  description="see manual"
variable: Translate_z double 74 3 description="see manual"

setGroup "UserInputs.Horizontal_STAB.Section_1"
variable: Xle1 double 79 1  description="see manual"
variable: Yle1 double 79 2  description="see manual"
variable: Zle1 double 79 3  description="see manual"
variable: Chord double 79 4  description="see manual"
variable: Elevator_Cgain double 83 2  description="see manual"
variable: Elevator_Xhinge  double 83 3  description="see manual"
variable: Elevator_SgnDup double 83 7  description="see manual"

setGroup "UserInputs.Horizontal_STAB.Section_2"
variable: Xle2 double 87 1  description="see manual"
variable: Yle2 double 87 2  description="see manual"
variable: Zle2 double 87 3  description="see manual"
variable: Chord double 87 4  description="see manual"
variable: Elevator_Cgain double 91 2  description="see manual"
variable: Elevator_Xhinge  double 91 3 description="see manual"
variable: Elevator_SgnDup double 91 7  description="see manual"

setGroup "UserInputs.Vertical_STAB.Translate"
variable: Translate_x double 99 1 description="see manual"
variable: Translate_y double 99 2  description="see manual"
variable: Translate_z double 99 3  description="see manual"

setGroup "UserInputs.Vertical_STAB.Section_1"
variable: Xle1 double 103 1  description="see manual"
variable: Yle1 double 103 2  description="see manual
variable: Zle1 double 103 3  description="see manual"
variable: Chord double 103 4 description="see manual"
variable: Rudder_Cgain double 107 2  description="see manual"
variable: Rudder_Xhinge  double 107 3  description="see manual"
variable: Rudder_SgnDup double 107 7  description="see manual"

setGroup "UserInputs.Vertical_STAB.Section_2"
```

```
variable: Xle2 double 111 1  description="see manual"
variable: Yle2 double 111 2  description="see manual"
variable: Zle2 double 111 3  description="see manual"
variable: Chord double 111 4  description="see manual"
variable: Rudder_Cgain double 115 2  description="see manual"
variable: Rudder_Xhinge  double 115 3 description="see manual"
variable: Rudder_SgnDup double 115 7 description="see manual"

}

RowFieldInputFile inputFile2
{
templateFile:  mtrn_MAV_b.template
fileToGenerate:  mtrn_MAV_b.batch

setDelimiters "= ,"
   setGroup UserInputs.Run_Constraints
variable: Roll_rate double 2 3  description="see manual"
variable: Pitch_rate double 3 3  description="see manual"
variable: Yaw_rate double 4 3  description="see manual"
variable: Flap double 5 3  description="see manual"
variable: Aileron double 6 3  description="see manual"
variable: Elevator double 7 3  description="see manual"
variable: Rudder double 8 3 description="see manual"
variable: AOA double 9 3  description="see manual"

}

RowFieldOutputFile outputFile
{
# This routine parses the program output file.
# Other variables can be extracted as desired.

 fileToParse: mtrn_MAV_results.txt

 setDelimiters "= ,"

 markAsBeginning "Configuration"
       setGroup Results.Run_Case_AERO_coeff
 variable: Cl_tot double 21 2 description="see manual"
 variable: Cd_tot double 22 2 description="see manual"
 variable: Cd_ind   double 23 4 description="see manual"
 variable: Cl_ff double 24 2 description="see manual"
 variable: Cd_ff double 24 4 description="see manual"
 variable: e double 25 4 description="see manual"

 setGroup Results.Conrol_surface_deflections
 variable: flap double 27 2 description="see manual"
 variable: aileron double 28 2 description="see manual"
 variable: elevator double 29 2 description="see manual"
 variable: rudder  double 30 2 description="see manual"

 setGroup Results.Stability_axis_derivatives
 markAsBeginning "CLa ="

 variable: CL_a double 1 6 description="see manual"
 variable: Cy_a double 2 6 description="see manual"
 variable: Cl_a double 3 5 description="see manual"
 variable: Cm_a double 4 6 description="see manual"
 variable: cn_a double 5 5 description="see manual"
 variable: CL_b double 1 8  description="see manual"
 variable: Cy_b double 2 8  description="see manual"
 variable: Cl_b double 3 7  description="see manual"
 variable: Cm_b double 4 8  description="see manual"
 variable: cn_b double 5 7  description="see manual"
```

```
 variable: Xnp double 27 4  description="neutral point"
}
```

Bibliography

1. "Blade MCX." http://www.horizonhobby.com/, Accessed 02.20.2009.
2. "Digital DATCOM+." http://www.holycows.net/datcom/, Accessed 01.20.2009.
3. "Hacker A10 9L Electric Motor." http://www.espritmodel.com/browseproducts/Hacker-A10-9L.HTML, Accessed 02.19.2009.
4. "KC's Power System Spreadsheet." http://www.rcgroups.com/ forums/showthread.php?t=232166, Accessed 02.04.2009.
5. "MH Airfoils." http://www.mh-aerotools.de/airfoils/, Accessed 01.28.2009.
6. "Properties of Air at Atmospheric Pressure in SI Units." http://www.engsolcom.com, Accessed 02.09.2009.
7. "Survey of Airflow Analysis Tools." http://gicl.cs.drexel.edu/wiki/Mongan, Accesed 01.28.2009.
8. Anderson, John Jr. *Hypersonic and High-Temperature Gas Dynamics* (2nd Edition). AIAA, 2006.
9. Barlow, Jewel B. and Rae, William H. and Pope, Alan. *Low Speed Wind Tunnel Testing* (3rd Edition). John Wiley & Sons Inc., 1999.
10. Barrett, R. *Developmental History of a New Family of Subscale, Convertible, High Performance UAVs*. Dept. of Aerospace Engineering, Delft University of Technology (Netherlands), 2003.
11. Beam, Jerry and Heidenreich, John and Miller, Ryan and Fellner, Joseph and Hager, Brian, "Development of a Distributed Integrated Modeling Environment to Study the Impact of Subsystem Performance on an Air Vehicle Design." *3rd International Energy Conversion Engineering Conference*, AIAA-2005-5573.
12. Becker, Jon, "Creating Vortex Lattice Models for the Piccolo Simulator with AVL." www.cloudcaptech.com/download/ Piccolo/Aircraft%20Modeling%20Tools/. Accessed 02.05.2009.
13. Blake, William B. and Simon, James M., "Tools For Rapid Analysis of Aircraft and Missile Aerodynamics," AIAA-1998-2793.
14. Coleman, Colin P., "A Survey of Theoretical and Experimental Coaxial Rotor Aerodynamic Research." *NASA Technical Paper-3675*, March 1997.
15. Deters, Robert W. and Selig, Michael S., "Static Testing of Micro Propellers." *26th AIAA Applied Aerodynamics Conference*, AIAA-2008-6246.

16. Dittmar, Josh E. *Integrated Conceptual Design of Joined-Wing Sensorcraft Using Response Surface Models*. MS thesis, Air Force Institute of Technology, WPAFB, Ohio, 2006.

17. Drela, Mark, "QPROP." www.web.mit.edu/drela/Public/web/qprop, Accessed 10.02.2008.

18. Drela, Mark, "QPROP Formulation." http://web.mit.edu/drela/ Public/web/qprop/qprop_theory.pdf. Accessed 2.12.2008.

19. Drela, Mark and Youngren, Harold, "Athena Vortex Lattice (AVL)." http://web.mit.edu/drela/Public/web/avl. Accessed 01.05.2009.

20. Galinski, Cezary, "Gust Resistant Fixed Wing Micro Air Vehicle." *Journal of Aircraft*, Vol.43, No.5, 2006. pp.1586-1589.

21. Grasmeyer, Joel M. and Keennon, Matthew T., "Development of the Black Widow Micro Air Vehicle." in *Fixed and Flapping Wing Aerodynamics for Micro Air Vehicle Applications* Ed. Thomas J. Mueller, Vol.195, Ch.24, AIAA 2001.

22. Green, William E. *A Multimodal Micro Air Vehicle for Autonomous Flight in Near Earth Environments*. PhD dissertation, Drexel University, May 2007.

23. Gur, Ohad and Rosen, Aviv. "*Comparison Between Blade Element Models of Propellers*." Unpublished Paper, Techion-Israel Institute of Technology, Haifa, 32000, Israel.

24. Gur, Ohad and Rosen, Aviv. "*Optimizing Electric Propulsion Systems for UAV's*." Unpublished Paper, Techion-Israel Institute of Technology, Haifa, 32000, Israel.

25. Hall, Alexander P. K. and Wong, K. C. and Auld, Doug, "Coaxial Aero-Mechanical Analysis of MAV Rotorcraft with Rotor Interaction for Optimisation." *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, AIAA-2008-6072.

26. He, Ruijie and Sato, Sho and Drela, Mark, "Design of a Single-Motor Nano Aerial Vehicle with a Gearless Torque-Canceling Mechanism." *46th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA-2008-1417.

27. Hogge, Jeffrey V. *Development of A Miniature VTOL Tail-Sitter Unmanned Aerial Vehicle*. MS thesis, Brigham Young University, Provo, Utah, 2008.

28. Huber, A. F. *Death by Thousand Cuts- Micro-Air Vehicles in the Service Of Air Force Missions*. 2001.

29. Hunsaker, D.F. *A Numerical Vortex Approach to Aerodynamic Modeling of SUAV/VTOL Aircraft*. MS thesis, Brigham Young University, Provo, Utah, 2006.

30. In, Hou and Jager, Leong and Jager, Rylan and Keshmiri, Shahriar and Colgren, Richard, "Development of a Pilot Training Platform for UAVs Using a 6DOF Nonlinear Model with Flight Test Validation." *AIAA Modeling and Simulation Technologies Conference and Exhibit*, AIAA-008-6368.

31. Jayaraman, Divahar, "4 Dimensional Visualization." www.mathworks.com /matlabcentral/fileexchange/13503. Accessed 09.05.2008.

32. Kang, Ning and Sun, Mao, "Simulated Flowfields in NearGround Operation of Single and Twin Rotor Configurations." *Journal of Aircraft*, Vol.37, No.2, 2000. pp.214-220.

33. Kellogg, James C., "Case Study." in *Introduction to The Design of Fixed-Wing Micro Air Vehicles, Including Three Case Studies* Ed. Joseph A. Schetz, Ch.4, AIAA 2006.

34. Kim, Hyo Won and Brown, Richard E, "Coaxial Rotor Performance and Wake Dynamics in Steady and Manoeuvring Flight." *American Helicopter Society 62nd Annual Forum*, 2006.

35. Krashanitsa, Roman and Platanitis, George and Silin, Dmytro and Shakarayev, Sergev, "Autopilot Integration into Micro Air Vehicles." in *Introduction to The Design of Fixed-Wing Micro Air Vehicles, Including Three Case Studies*, 2006.

36. Lamar, John E., "Predictions of Vortex Flow Characteristics of Wings at Subsonic and Supersonic Speeds." *Journal of Aircraft*, Vol.13, No.7, 1976. pp.440-494.

37. Lupo, Salvatore and Nyberg, Henrik and Karlsson, Anders and Mohseni, Kamran, "Xwing A 3D Viscous Design Tool for Wings." *46th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA-2008-173.

38. Marek, Przemyslaw. *Design, Optimization and Flight Testing of A Micro Air Vehicle*. MS thesis, University of Glasgow, December 2007.

39. McDonnel Douglas Astronautics Company St. Louis Division, St.Louis, Missouri 63166. *The USAF Stability and Control Digital Datcom*, April 1979.

40. McMichael, J.M. and Francis, M.S. *Micro Air Vehicles Toward a New Dimension in Flight*. 1997.

41. Melin, Tomas. *A Vortex Lattice MATLAB Implementation for Linear Aerodynamic Wing Applications*. MS thesis, Royal Institute of Technology (KTH), Stockholm, Sweden, December 2000.

42. Melin, Tomas. *User's Guide, Reference Manual, Tornado 1.0, Release 2.3*. Royal Institute of Technology (KTH), Stockholm, Sweden, 2001.

43. Moschetta, Jean-Marc and Bataill, Boris and Thipyopas, Chinnapat and Shkarayev, Sergey, "On Fixed-Wing Micro-Air Vehicles with Hovering Capabilities." *46th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA-2008-221.

44. Mueller, Thomas J., "Aerodynamic Measurements at Low Reynolds Numbers for Fixed Wing Micro Air Vehicles." *Development and Operation of UAVs for Military and Civilian Applications* , RTO AVT/Special Course 1999.

45. Mueller, Thomas J., editor. *Fixed and Flapping Wing Aerodynamics for Micro Air Vehicles*, *195*. Reston, VA: AIAA, 2001.

46. Mueller, Thomas J. and Kellogg, James C. and Ifju, Peter G. and Shkarayev, Sergey V. *Introduction to The Design of Fixed-Wing Micro Air Vehicles, Including Three Case Studies*. AIAA Education Series, 2007.

47. Mueller, Thomas J. and Torres, Gabriel E. *Aerodynamics of Low Aspect Ratio Wings at Low Reynolds Numbers with Application to Micro Air Vehicle Design and Optimization*. Technical Report, University of Notre Dame, November 2001.

48. Mueller, Thomas J. and Torres, Gabriel E., "Elements of Aerodynamics, Propulsion and Design." in *Introduction to The Design of Fixed-Wing Micro Air Vehicles, Including Three Case Studies* Ed. Joseph A. Schetz, Ch.2, AIAA 2006.

49. Null, William and Noscek, Aaron and Shkarayev, Sergey, "Effects of Propulsive-Induced Flow on the Aerodynamics of Micro Air Vehicles." *23rd AIAA Applied Aerodynamics Conference*, AIAA-2005-4616.

50. PhoenixIntegration(PHX). www.phoenix-int.com, Accessed 05.05.2008.

51. Pines, Darryll J. and Bohorquez, Felipe, "Challenges Facing Future Micro-Air-Vehicle Development." *Journal of Aircraft*, Vol.43, No.2, 2006. pp.290-305.

52. Polhamus, Edward C., "Predictions of Vortex-Lift Characteristics by a Leading-Edge-Suction Analogy." *Journal of Aircraft*, Vol.34, No.4, 1971. pp.193-199.

53. Profili2. www.profili2.com, Accessed 11.20.2008.

54. Raymer, Daniel P. *Aircraft Design: A Conceptual Approach-4th Edition*. AIAA, 2006.

55. Selig, Michael S. and Gopalarathnam, Ashok and Giguere, Philippe and Lyon, Christopher A., "Systematic Airfoil Design Studies at Low Reynolds Numbers." in *Fixed and Flapping Wing Aerodynamics for Micro Air Vehicle Applications* Ed. Thomas J. Mueller, Vol.195, Ch.8, AIAA 2001.

56. Sibilski, Krzysztof and Pietrucha, Jozef and Zlocka, Maria, "The Comparative Evaluation of Power Requirements for Fixed, Rotary, and Flapping Wings Micro Air Vehicles." *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, AIAA-2007-6498.

57. Stanford, Bret and Abdulrahim, Mujahid and Lind, Rick and Ifju, Peter, "Investigation of Membrane Actuation for Roll Control of a Micro Air Vehicle." *Journal of Aircraft*, Vol.44, No.3, 2007. pp.741-749.

58. Stewart, Kelly and Abate, Gregg and Evers, Johnny, “Flight Mechanics and Control Issues for Micro Air Vehicles.” *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, AIAA-2006-6638.

59. Torres, Gabriel E. *Aerodynamics of Low Aspect Ratio Wings at Low Reynolds Numbers with Application to Micro-Air-Vehicle Design*. PhD dissertation, University of Notre Dame, 2002.

60. Torres, Gabriel E. and Mueller, Thomas J., “Aerodynamic Characteristics of Low Aspect Ratio Wings at Low Reynolds Numbers.” in *Fixed and Flapping Wing Aerodynamics for Micro Air Vehicle Applications* Ed. Thomas J. Mueller, Vol.195, Ch.7, AIAA 2001.

61. van Sommeren, Ron, “Eflight Calculators.” http://www.rcgroups.com /forums /showthread.php?t=606703, Accessed 09.20.2008.

Vita

1st Lt.Turan was born in Burdur, TURKIYE (Turkey). He graduated from Kuleli Military High School in 1998 and from the Turkish Air Force Academy in 2002 receiving his Bachelor's Degree in Aeronautical and Astronautical Engineering. He started Undergraduate Pilot Training (UPT) at 2nd Main Jet Base, Cigli, Izmir. After being selected for Euro Nato Joint Jet Pilot Training (ENNJPT), he started a two-month language program at the Defense Language Institute (DLI) at Lackland AFB, San Antonio, Texas in 2003 and resumed UPT at Sheppard AFB, Wichita Falls, Texas and graduated in July 2004. He is an F-4E Phantom 2020 pilot and has continued flying in the front and back seat until coming to the Air Force Institute of Technology (AFIT) in 2007.

Following the completion of his Master's Degree, 1st Lt. Turan will attend the Turkish Air War College in Istanbul.

Permanent address: 2950 Hobson Way
Air Force Institute of Technology
Wright-Patterson AFB, OH 45433

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE (DD–MM–YYYY)	2. REPORT TYPE	3. DATES COVERED (*From — To*)
26-03-2009	Master's Thesis	Sept 2007- Mar 2009

4. TITLE AND SUBTITLE

Tools For The Conceptual Design
And
Engineering Analysis Of Micro Air Vehicles

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Mustafa Turan, 1st Lieutenant, (TUAF)

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/ENY)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GAE/ENY/09-M19

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Labs (AFRL/RBAA)

Attn: Dr. Michael V. OL
2130 8th St., Bldg 45
Wright-Patterson AFB, OH 45433-7542
(937) 255-1196 fax (937) 656-7868

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Micro Air Vehicles (MAV) are a subset of Unmanned Aircraft (UAS) that are up to two orders of magnitude smaller than manned systems. Near-Earth environments, such as forests, caves, tunnels and urban structures make reconnaissance, surveillance and search-and-rescue missions difficult and dangerous to accomplish. Therefore, MAVs are considered ideal for these types of missions. However, the data using full size aircraft is inadequate to characterize miniature aircraft parameters due to the lower Reynolds numbers and low aspect ratio (LAR) wings and impact of wing-propeller interactions. The main objectives of this research were to: collect and synthesize the available data/tools; create a statistically integrated database/tool set of MAV designs for conceptual design trades; validate the tool set using published experimental data; synthesize and model a prototype design using conceptual and empirical analysis; highlight MAV-specific design criteria and identify gaps in existing data for later research. The following design tools have constituted the starting point for creating a demonstration tool-set for MAV design: Digital DATCOM (aerodynamics), Athena Vortex Lattice (AVL) (stability and control), QPROP (propeller, motor, and energy requirement), MATLAB (various applications), Microsoft Excel (power/battery modeling) and Phoenix Integration ModelCenter (MC) as the executive control program (integration, sizing and trade studies). Validation cases were completed for the current level of the single-prop, fixed-wing design tool. A coaxial MAV prototype was evaluated and some parametric studies were conducted for QPROP performance.

15. SUBJECT TERMS
Micro Air Vehicle, Low Re Numbers, DATCOM, QPROP, Athena Vortex Lattice, ModelCenter

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
					Frederick G. Harmon, Lt Col, (USAF)
a. REPORT	b. ABSTRACT	c. THIS PAGE	UU	156	19b. TELEPHONE NUMBER (Include Area Code)
U	U	U			(937) 255-3636, ext 7478;frederick.harmon@ afit.edu